目录

第 1 章　小波分析基础理论

小波分析（Wavelet Analysis）是在现代调和分析的基础上发展起来的一门新兴学科，其基础理论知识涉及到泛函分析、傅里叶分析、信号与系统、数字信号处理等诸方面，同时具有理论深刻和应用十分广泛双重意义。这里我们只准备对小波分析的整体思想进行介绍，不进行相关的详细数学推导和证明，以让大多数读者尽快对小波分析有一个感性认识，并能初步应用小波分析来解决工程中的实际问题。如果读者想在此基础上进一步了解小波分析的理论背景，可参考相关的资料。

本章主要内容：

- MATLAB 6.5 语言简介
- 小波变换的由来
- 小波变换定义及特点
- 一维连续小波变换
- 离散小波变换
- 小波包分析
- 常用小波函数

1.1　MATLAB 6.5 语言简介

1.1.1　MATLAB 的产生背景及主要产品

MATLAB 诞生于 20 世纪 70 年代，它的编写者是 Cleve Moler 博士和他的同事。当时，Cleve Moler 博士和他的同事开发了 EISPACK 和 LINPACK 的 FORTRAN 子程序库。这两个程序库主要是求解线性方程的程序库。但是，Cleve Moler 发现学生使用这两个程序库时有困难，主要是接口程序不好写，很费时间。于是 Cleve Moler 自己动手，在业余时间里编写了 EISPACK 和 LINPACK 的接口程序。Cleve Moler 给这个接口程序取名为 MATLAB，意为矩阵（Matrix）和实验室（Laboratory）的组合。以后几年，MATLAB 作为免费软件在大学里使用，深受大学生的喜爱。

1984 年，Cleve Moler 和 John Little 成立了 MathWorks 公司，正式把 MATLAB 推向市场，并继续进行 MATLAB 的开发。1993 年，MathWorks 公司推出 MATLAB 4.0；1995 年，MathWorks 公司推出 MATLAB 4.2C 版（For Win3.x）；1997 年推出 MATLAB 5.0；2000 年 10 月，MathWorks 公司推出 MATLAB 6.0；2002 年 8 月，新的版本 MATLAB 6.5 已经开始发布了。每一次版本的推出都使 MATLAB 有了长足的进步，界面越来越友好，内容越来越丰富，功能越来越强大。它的帮助信息采用超文本格式和 PDF 格式，可以很方便

地阅读。

MATLAB 长于数值计算，能处理大量的数据，而且效率比较高。MathWorks 公司在此基础上开拓了符号计算、文字处理、可视化建模和实时控制能力，增强了 MATLAB 的市场竞争力，使 MATLAB 成为市场上流的数值计算软件。

MATLAB 产品组是支持从概念设计、算法开发、建模仿真，到实时实现的理想的集成环境。无论是进行科学研究和产品开发，MATLAB 产品组都是必不可少的工具。

MATLAB 产品组可用来进行：

- 数据分析
- 数值和符号计算
- 工程与科学绘图
- 控制系统设计
- 数字图像信号处理
- 财务工程
- 建模、仿真、原型开发
- 应用开发
- 图形用户界面设计

MATLAB 产品组被广泛地应用于包括信号与图像处理、控制系统设计、通信、系统仿真等诸多领域。开放式的结构使 MATLAB 产品组很容易针对特定的需求进行扩充，从而在不断深化对问题认识的同时，提高自身的竞争力。

MATLAB 产品组的一大特性是有众多的面向具体应用的工具箱和仿真块，包含了完整的函数集用来对信号图像处理、控制系统设计、神经网络等特殊应用进行分析和设计。其他的产品延伸了 MATLAB 的能力，包括数据采集、报告生成和依靠 MATLAB 语言编程产生独立 C/C++代码等等。

MATLAB 主要产品构成：

（1）MATLAB

所有 MathWorks 公司产品的数值分析和图形基础环境。MATLAB 将 2D 和 3D 图形、MATLAB 语言能力集成到一个单一的，易学易用的环境之中。

（2）MATLAB Toolbox

一系列专用的 MATLAB 函数库，解决特定领域的问题。工具箱是开放的可扩展的，可以查看其中的算法，或开发自己的算法。

（3）MATLAB Compiler

将 MATLAB 语言编写的 M 文件自动转换成 C 或 C++文件，支持用户进行独立应用开发。结合 MathWorks 提供的 C/C++数学库和图形库，用户可以利用 MATLAB 快速地开发出功能强大的独立应用。

（4）Simulink

结合了框图界面和交互仿真能力的非线性动态系统仿真工具。它以 MATLAB 的核心数学、图形和语言为基础。

（5）Stateflow

与 Simulink 框图模型相结合，描述复杂事件驱动系统的逻辑行为，驱动系统在不同

的模式之间进行切换。

（6）Real-Time Workshop

直接从 Simulink 框图自动生成 C 或 Ada 代码，用于快速原型和硬件在回路仿真，整个代码生成可以根据需要完全定制。

（7）Simulink Blockset

专门为特定领域设计的 Simulink 功能块的集合，用户也可以利用已有的块或自编写的 C 和 MATLAB 程序建立自己的块

1.1.2 MATLAB 语言特点

MATLAB 语言有不同于其他高级语言的特点，它被称为第四代计算机语言。正如第三代计算机语言如 FORTRAN 语言与 C 语言等使人们摆脱了对计算机硬件的操作一样，MATLAB 语言使人们从繁琐的程序代码中解放出来。它的丰富的函数使开发者无需重复编程，只要简单地调用和使用。MATLAB 语言最大的特点是简单和直接。MATLAB 语言的主要特点有：

1．编程效率高

MATLAB 是一种面向科学与工程计算的高级语言，允许用数学形式的语言编写程序，且比 BASIC、FORTRAN 和 C 等语言更加接近我们书写计算公式的思维方式，用 MATLAB 编写程序犹如在演算纸上排列出公式与求解问题。因此，MATLAB 语言也可通俗地称为演算纸式科学算法语言。由于它编写简单，所以编程效率高，易学易懂。

2．用户使用方便

MATLAB 语言是一种解释执行的语言（在没被专门的工具编译之前），它灵活、方便，其调试程序手段丰富，调试速度快，需要学习时间少。人们用任何一种语言编写程序和调试程序一般都要经过四个步骤：编辑、编译、连接，以及执行和调试。各个步骤之间是顺序关系，编程的过程就是在它们之间做瀑布型的循环。MATLAB 语言与其他语言相比，较好地解决了上述问题，把编辑、编译、连接和执行融为一体。它能在同一画面上进行灵活操作，快速排除输入程序中的书写错误、语法错误，以至语意错误，从而加快了用户编写、修改和调试程序的速度，可以说在编程和调试过程中它是一种比 VB 还要简单的语言。

具体地说，MATLAB 运行时，如直接在命令行输入 MATLAB 语句（命令），包括调用 M 文件的语句，每输入一条语句，就立即对其进行处理，完成编译、连接和运行的全过程。又如，将 MATLAB 源程序编辑为 M 文件，由于 MATLAB 磁盘文件也是 M 文件，所以编辑后的源文件就可直接运行，而不需进行编译和连接。在运行 M 文件时，如果有错，计算机会在屏幕上给出详细的出错信息，用户经修改后再执行，直到正确为止。所以可以说，MATLAB 语言不仅是一种语言，广义上讲是一种语言开发系统，即语言调试系统。

3．扩充能力强，交互性好

高版本的 MATLAB 语言有丰富的库函数，在进行复杂的数学运算时可以直接调用，而且 MATLAB 的库函数同用户文件在形式上一样，所以用户文件也可作为 MATLAB 的

库函数来调用。因而，用户可以根据自己的需要方便地建立和扩充新的库函数，以便提高MATLAB 使用效率和扩充它的功能。另外，为了充分利用 FORTRAN、C 等语言的资源，包括用户已编好的 FORTRAN、C 语言程序，通过建立 Me 调文件的形式，混合编程，方便地调用有关的 FORTRAN、C 语言的子程序，还可以在 C 语言和 FORTRAN 语言中方便地使用 MATLAB 的数值计算功能。这样良好的交互性使程序员可以使用以前编写过的程序，减少重复性工作，也使现在编写的程序具有重复利用的价值。

4．移植性很好，开放性很好

MATLAB 是用 C 语言编写的，而 C 语言的可移植性很好。于是 MATLAB 可以很方便地移植到能运行 C 语言的操作平台上。MATLAB 适合的工作平台有：Windows 系列、UNIX、Linux、VMS6.1、PowerMac。除了内部函数外，MATLAB 所有的核心文件和工具箱文件都是公开的，都是可读可写的源文件，用户可以通过对源文件的修改和自己编程构成新的工具箱。

5．语句简单，内涵丰富

MATLAB 语言中最基本最重要的成分是函数，其一般形式为[*a*，*b*，*c*，…]= fun（*d*，*e*，*f*，…），即一个函数由函数名、输入变量（*d*，*e*，*f*，…）和输出变量（*a*，*b*，*c*，…）组成，同一函数名 F，不同数目的输入变量（包括无输入变量）及不同数目的输出变量，代表着不同的含义（有点像面向对象中的多态性）。这不仅使 MATLAB 的库函数功能更丰富，而且还大大减小了需要的磁盘空间，使得 MATLAB 编写的 M 文件简单、短小而高效。

6．高效方便的矩阵和数组运算

MATLAB 语言像 BASIC、FORTRAN 和 C 语言一样规定了矩阵的算术运算符、关系运算符、逻辑运算符、条件运算符及赋值运算符，而且这些运算符大部分可以毫无改变地照搬到数组间的运算，有些如算术运算符只要增加“·”就可用于数组间的运算。另外，它不需定义数组的维数，并给出矩阵函数、特殊矩阵专门的库函数，使之在求解诸如信号处理、建模、系统识别、控制、优化等领域的问题时，显得大为简捷、高效、方便，这是其他高级语言所不能比拟的。在此基础上，高版本的 MATLAB 已逐步扩展到科学及工程计算的其他领域。因此，不久的将来，它一定能名副其实地成为“万能演算纸式的”科学算法语言。

7．方便的绘图功能

MATLAB 的绘图是十分方便的，它有一系列绘图函数（命令），例如线性坐标、对数坐标、半对数坐标及极坐标，均只需调用不同的绘图函数（命令），在图上标出图题、*XY* 轴标注，格（栅）绘制也只需调用相应的命令，简单易行。另外，在调用绘图函数时调整自变量可绘出不变颜色的点、线、复线或多重线。这种为科学研究着想的设计是通用的编程语言所不及的。

1.1.3 MATLAB 6.5 新特点

2002 年，MathWorks 公司发布 MATLAB 最新 6.5 版产品。MATLAB 6.5 的特点在于

全新的桌面及各种不同领域的集成工具，使用户易于使用。多种新工具简化了一般的工作，如资料输入、快速分析，并创造高品质且具实用性的图表分析等。MATLAB 6.5 包含了新的 JIT 加速度计，JIT 加速度计有力地增加了 MATLAB 中的许多操作和数据类型的计算速度。其他新的特色和加强包括以下三个方面：

1．编程和数据类型

（1）增加了变量名、函数名和文件名的最大长度：变量名、函数名、子函数名、结构域名、M 文件名、MEX 文件名和 MDL 文件名可以达到 63 字节；

（2）支持 64 位的文件偏移量，能够为大于 2G 字节的数据文件实现低层次的 I/O 函数；

（3）支持有符号和无符号的 64 位整数；

（4）支持使用动态域名来访问和修改结构数组；

（5）简化了 AND 和 OR 逻辑运算；

（6）支持新的 MATLAB 定时器，而不是定时执行 MATLAB 命令；

（7）改进了音频支持；

（8）加强了警告和错误提示功能，支持格式化的字符串和消息标识符。

2．外部接口

（1）改进了自动化客户接口：新的查看和修改属性用户接口，增强了事件和例外句柄；

（2）增强了网络集成：读 URL 的内容，在 MATLAB 中发送 E-mail 以及解压缩文件。

3．开发环境

（1）新的 M-文件接口，能更好地理解 M-代码；

（2）新的启动按钮，易于执行共同的命令；

（3）改进了文件和目录管理工具；

（4）增强了数组编辑器：与 Excel 之间剪切、复制、删除和交换单元的新功能，支持更大的数组；

（5）改进了编辑和调试工具；

（6）改进与 PC 平台的控制接口；

（7）支持新的图形用户接口，从 HDF 或 HDF-EOS 文件导入数据；

（8）JVM1.3.1 支持 Windows、Linux 和 Solaris 平台。

4．图形

提高了图形性能：新的彩色图形编辑器；改进了图形特性编辑器。

5．数学

新的数学计算和算法改进：在 Pentium 4 上更快地计算许多函数，比如矩阵乘法、矩阵转置和线性代数运算。特别指出的是，许多新特色并不是适合于所有的平台。

1.2 小波变换的由来

在我们的周围，每天都有大量的信号需要我们进行分析，例如我们说话的声音、机器的振动、金融变化数据、地震信号、音乐信号、医疗图像等。相当多的信号需要进行有效的编码、压缩、消噪、重建、建模和特征提取。因此，人们一直在努力寻找各种有效的信号处理方法。

众所周知，自从 1822 年傅里叶（Fourier）发表“热传导解析理论”以来，傅里叶变换一直是信号处理领域中应用最广泛的一种分析手段。傅里叶变换的基本思想是将信号分解成一系列不同频率的连续正弦波的叠加，或者从另外一个角度来说是将信号从时间域转换到频率域，如图 1-1 所示。对于许多情况，傅里叶分析是能够很好地满足分析要求的。

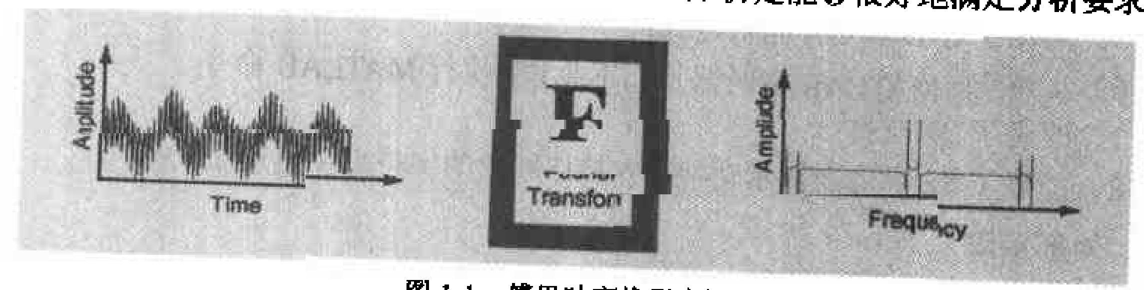


图 1-1　傅里叶变换示意图

但是傅里叶变换有一个严重的不足，那就是在做变换时丢掉了时间信息，无法根据傅里叶变换的结果判断一个特定的信号是在什么时候发生的。也就是说，傅里叶变换只是一种纯频域的分析方法，它在频域里的定位是完全准确的（即频域分辨率最高），而在时域无任何定位性（或无分辨能力）。

如果要分析的信号是一种平稳信号，这一点也许不是很重要。然而实际中，大多数信号均含有大量的非稳态成分，例如偏移、趋势、突变、事件的起始与终止等情况，而这些情况往往是相当重要的，反映了信号的重要特征。例如，常见的音乐信号，在不同时间演奏不同音符；语音信号，在不同时间对应不同音节；探地信号，在目标出现的位置对应一个回波信号等。它们的频域特性都随时间而变化。对这一类时变信号进行分析，通常需要提取某一时间段（或瞬间）的频域信息或某一频率段所对应的时间信息。因此，需要寻求一种具有一定的时间和频率分辨率的基函数来分析时变信号。

为了研究信号在局部时间范围的频域特征，1946 年 Gabor 提出了著名的 Gabor 变换，之后进一步发展成为短时傅里叶变换（Short Time Fourier Transform，简称 STFT，又称为加窗傅里叶变换）。其基本思路是给信号加一个小窗，信号的傅里叶变换主要集中在对小窗内的信号进行变换，因此可以反映出信号的局部特征，如图 1-2 所示。目前，STFT 在许多领域获得了广泛的应用。但由于 STFT 的定义决定了其窗函数的大小和形状均与时间和频率无关而保持固定不变，这对于分析时变信号来说是不利的。高频信号一般持续时间很短，而低频信号持续时间较长。因此，我们期望对于高频信号采用小时间窗，对于低频信号则采用大时间窗进行分析。在进行信号分析时，这种变时间窗的要求同 STFT 的固定时窗（窗不随频率而变化）的特性是相矛盾的。这表明 STFT 在处理这一类问题时已无能

为力了。此外，在进行数值计算时，人们希望将基函数离散化，以节约计算时间和存储量。但 Gabor 基无论怎样离散，都不能构成一组正交基，因而给数值计算带来不便。

以上 Gabor 变换的不足之处，恰恰是小波变换的特长所在。小波变换不但继承和发展了 STFT 的局部化思想，而且克服了窗口大小不随频率变化、缺乏离散正交基的缺点，是一种比较理想的信号处理方法。图 1-3 是各种分析信号方法的对比示意图。

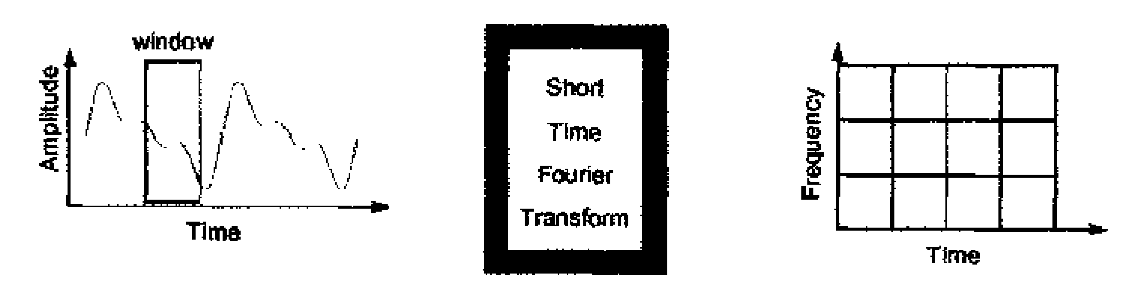


图 1-2　短时傅里叶变换示意图

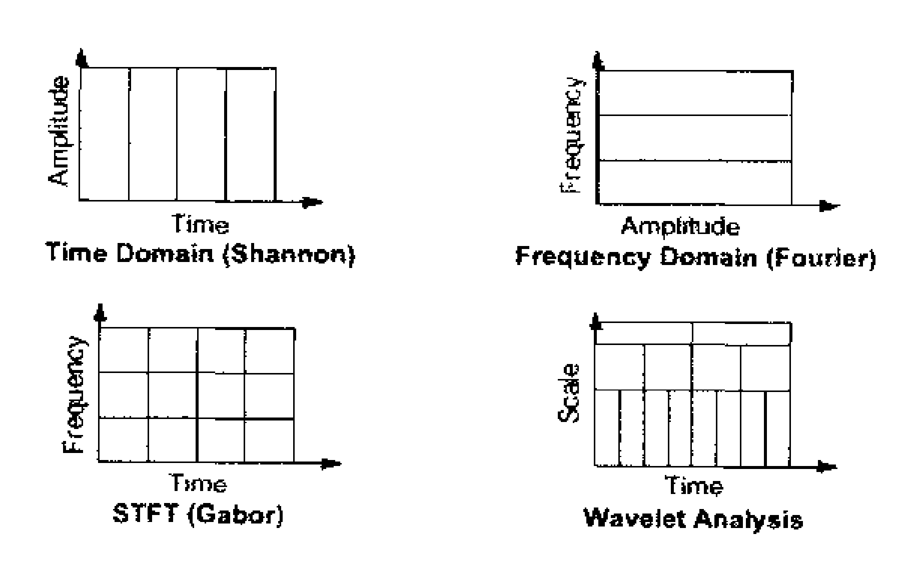
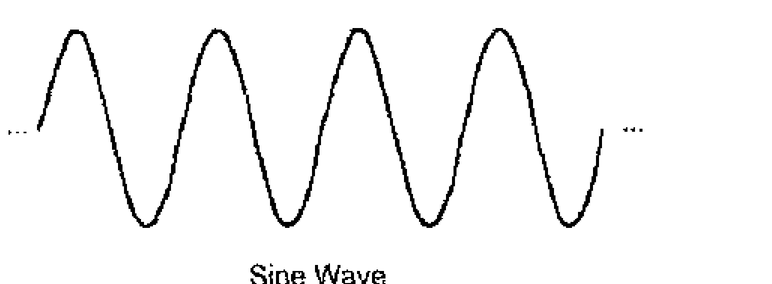


图 1-3　各种信号分析方法的对比

1.3　小波变换定义及特点

小波（wavelet），即小区域的波，是一种特殊的长度有限、平均值为0的波形。它有两个特点：一是“小”，即在时域都具有紧支集或近似紧支集；二是正负交替的“波动性”，也即直流分量为零。我们可以用小波和构成傅里叶分析基础的正弦波做一个比较，如图1-4所示。傅里叶分析所用的正弦波在时间上没有限制，从负无穷到正无穷，但小波倾向于不规则与不对称。傅里叶分析是将信号分解成一系列不同频率的正弦波的叠加，同样小波分析是将信号分解成一系列小波函数的叠加，而这些小波函数都是由一个母小波函数经过平移与尺度伸缩得来的。根据直觉，用不规则的小波函数来逼近尖锐变化的信号显然要比光滑的正弦曲线要好，同样，信号局部的特性用小波函数来逼近显然要比光滑的正弦函数来逼近要好。这里讨论的是一维的情况，小波分析同样可以用于二维图形的分析。



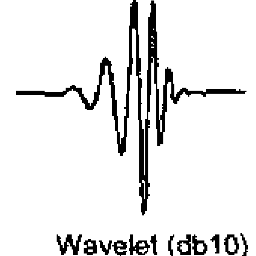


图 1-4　小波与傅里叶正弦波

小波变换的定义是把某一被称为基本小波（也叫母小波 mother wavelet）的函数 $\psi(t)$ 做位移 τ 后，再在不同尺度 a 下与待分析的信号 $x(t)$ 做内积：

$$WT_x(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}x(t)\psi^*(\frac{t-\tau}{a})\mathrm{d}t\,,\ a>0$$

等效的频域表示是：

$$WT_x(a,\tau)=\frac{\sqrt{a}}{2\pi}\int_{-\infty}^{\infty}X(\omega)\Psi^*(a\omega)e^{+j\omega\tau}\mathrm{d}\omega$$

式中 $X(\omega)$ 和 $\Psi(\omega)$ 分别是 $x(t)$ 和 $\psi(t)$ 的傅里叶变换。

可以这样理解上面表达式的意义：打个比喻，我们用镜头观察目标 $x(t)$（即待分析信号），$\psi(t)$ 代表镜头所起的作用（例如滤波或卷积）。τ 相当于使镜头相对于目标平行移动，a 的作用相当于镜头向目标推进或远离。由此可见小波变换有以下特点：

- 有多分辨率（multi-resolution），也叫多尺度（multi-scale）的特点，可以由粗及细地逐步观察信号。
- 可以看成用基本频率特性为 $\Psi(\omega)$ 的带通滤波器在不同尺度 a 下对信号做滤波。由于傅里叶变换的尺度特性可知这组滤波器具有品质因数恒定，即相对带宽（带宽与中心频率之比）恒定的特点。注意，a 越大相当频率越低。
- 适当地选择基小波，使 $\psi(t)$ 在时域上为有限支撑，$\Psi(\omega)$ 在频域上也比较集中，就可以使 WT 在时、频域都具有表征信号局部特征的能力，因此有利于检测信号的瞬态或奇异点。

正是由于上述特性，有人把小波变换誉为分析信号的数学显微镜。

如上所述，小波分析的一个主要优点就是能够分析信号的局部特征，例如可以发现叠加在一个非常规范的正弦信号上的一个非常小的畸变信号的出现时间，如图1-5所示。传统的傅里叶变换只能得到如图1-1所示的图形，为平坦的频谱上的两个尖峰。利用小波分析可以非常准确地分析出信号在什么时刻发生畸变。小波分析可以检测出许多其他分析方法忽略的信号特性，例如，信号的趋势，信号的高阶不连续点、自相似特性。小波分析还能以非常小的失真度实现对信号的压缩与消噪，它在图像数据压缩方面的潜力已经得到确认。在二维情况下，小波分析除了“显微”能力外还具有“极化”能力（即方向选择性），因而引人注意。

总之，小波变换作为一种数学理论和方法在科学技术和工程界引起了越来越多的关注

和重视。尤其在工程应用领域，特别是在信号处理、图像处理、模式识别、语音识别、量子物理、地震勘测、流体力学、电磁场、CT成像、机器视觉、机械状态监控与故障诊断、分形、数值计算等领域被认为是近年来在工具和方法上的重大突破。

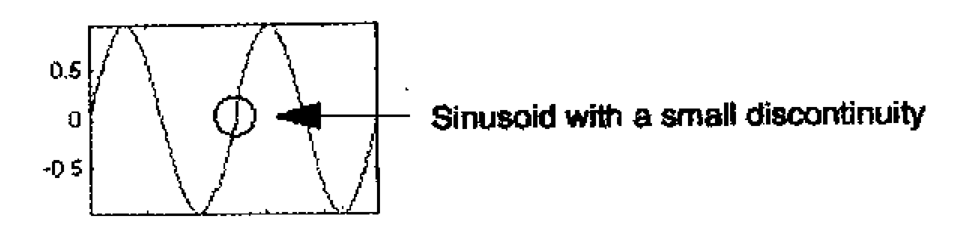


图 1-5　畸变信号

1.4　一维连续小波变换

前面我们介绍了小波变换的定义和特点，为让读者对小波变换有初步的感性认识，从本节开始介绍几种常见的小波变换和常用的小波函数。

傅里叶变换的数学表达式是：

$$F(\omega)=\int_{-\infty}^{+\infty} f(t)e^{-j\omega t}\mathrm{d}t$$

实际上是求$f(t)$在函数$e^{-j\omega t}$的投影值是多少，变换的结果为傅里叶系数，将这些系数乘上$e^{j\omega t}$，然后叠加起来，就得到了原来的信号值。其直观形式如图1-6所示。

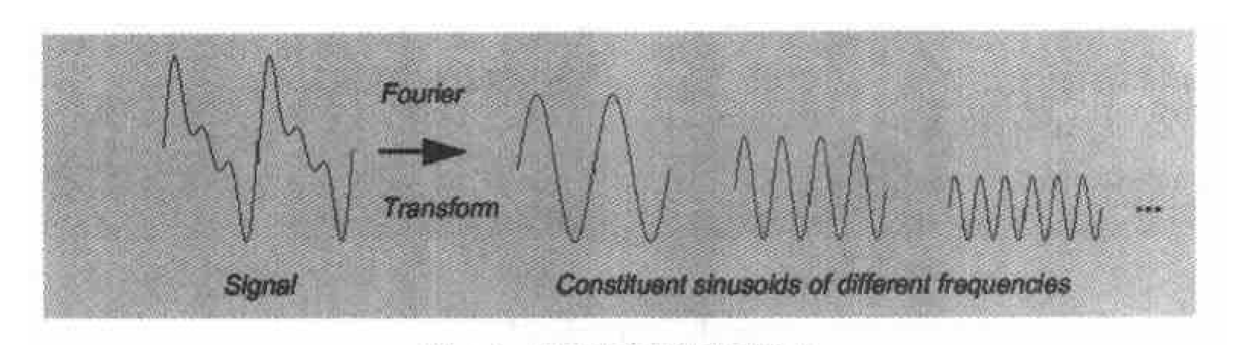


图 1-6　傅里叶变换的直观形式

同样小波变换是求$f(t)$在各个小波函数上的投影值是多少，由于每个小波函数均由个母小波函数经过尺度伸缩a与时间平移τ得来的。因此，小波变换可以写成如下形式：

$$C(scale,\ time_postion)=\int f(\tau)\Psi(scale,\tau)\mathrm{d}\tau$$

将上面得到的每一个系数同相应经过伸缩和平移后的小波函数相乘并叠加就可以恢复出原始信号，如图 1-7 所示。

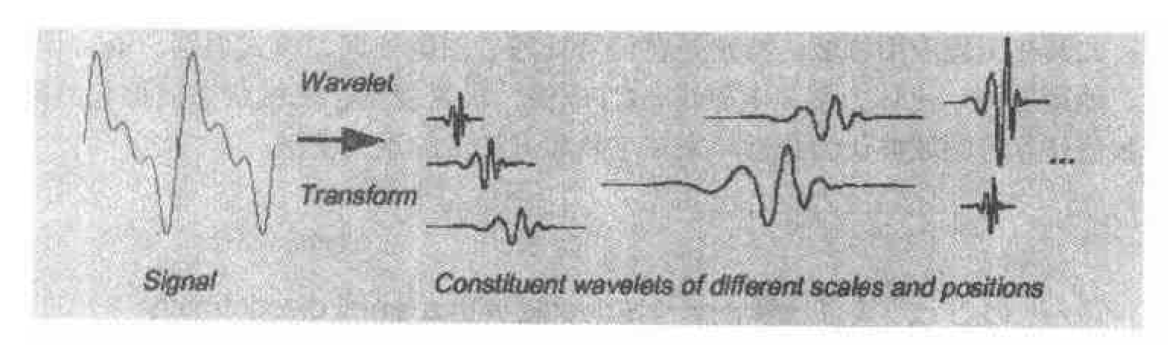


图 1-7 小波变换

应该尽量用一个函数来生成所有的基函数，不要用太多的函数，因为这样比较简单。这些生成的小波基函数最后两两正交。信号分析的一般思路就是分解与组合，寻找一组最能代表信号特征的函数形式，将信号用这些量来逼近，或者写成这些量的线性组合的形式。

小波分析也可以用滤波器的观点描述。高分辨率相当于应用高通滤波器，低分辨率相当于用低通滤波器来分析信号。

小波变换的思想来源于伸缩和平移方法，下面我们对其含义进行说明。

1. **尺度伸缩（Scaling）**

对波形的尺度伸缩就是在时间轴上对信号进行压缩与伸展，例如对于正弦波，有如图1-8 (a)所示的压缩，同样对于小波，有类似的尺度伸缩（如图1-8(b)所示）。可见，时间尺度a反比于频率。

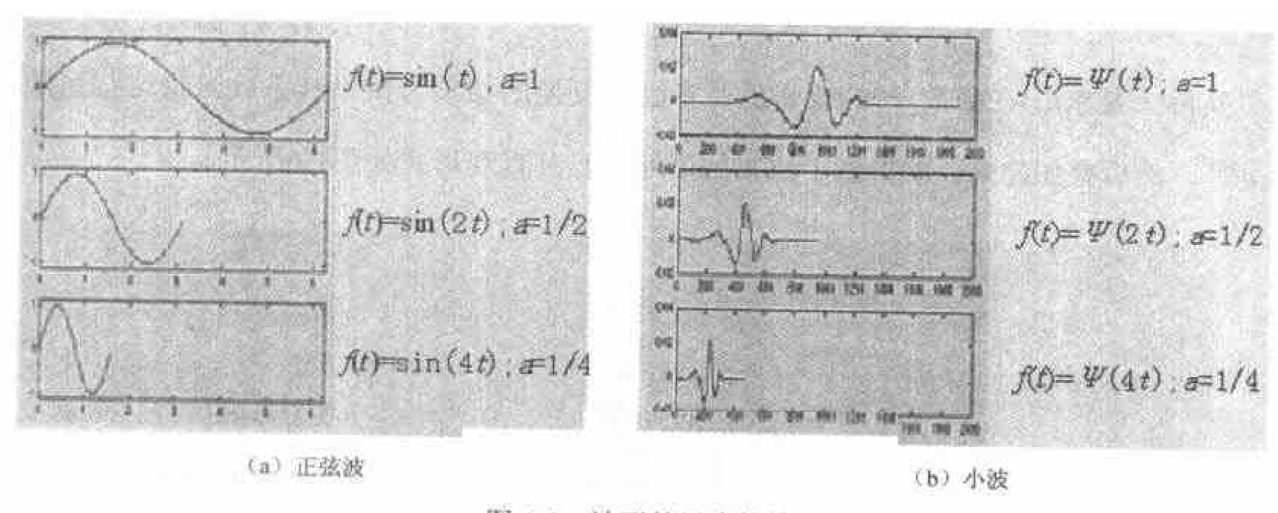


图 1-8 波形的尺度伸缩

2. **时间平移（Shifting）**

时间平移就是指小波函数在时间轴上的波形平行移动，如图1-9所示。



图 1-9 时间平移

下面介绍进行直观的连续小波运算的5个基本步骤：

（1）选择一个小波函数，并将这个小波与要分析的信号起始点对齐；

（2）计算在这一时刻要分析的信号与小波函数的逼近程度，亦即计算小波变换系数C，C越大，就意味着此刻信号与所选择的小波函数波形越相像（如图1-10所示）；

（3）将小波函数沿着时间轴向右移动一个单位时间，然后重复步骤（1）、（2）求出此时的小波变换系数C，直到覆盖完整个信号长度（如图1-11所示）；

（4）将所选择的小波函数尺度伸缩一个单位，然后重复步骤（1）、（2）、（3）（如图1-12所示）；

（5）对所有的尺度重复步骤（1）、（2）、（3）、（4）。

进行完上述步骤，将得到使用不同的尺度评估信号在不同的时间段的大量的系数。这些系数表征了原始信号在这些小波函数上的投影大小。可以以图形的方式直观地展示计算得到的结果。

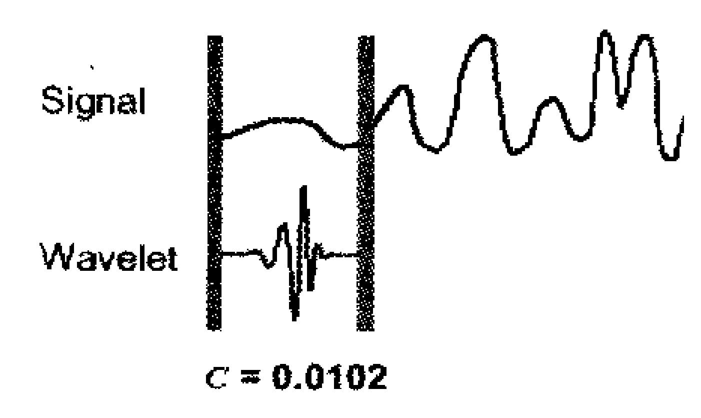


图1-10 小波运算1

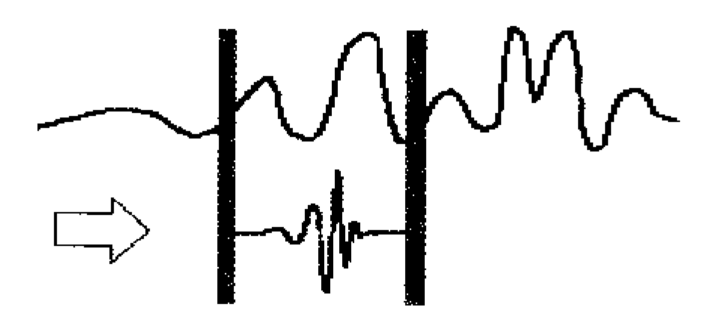


图1-11 小波运算2



图1-12 小波运算3

3. 尺度与频率的关系

尺度越大，意味着小波函数在时间上越长，亦即被分析的信号区间也就越长，因此，尺度越大意味着频率的分辨率就越低，主要获取的是信号的低频特性。反之，尺度越小，意味着只与信号的非常小的局部进行比较，因此主要获取是系统的高频特性。因此，它们之间的关系可以归纳如下：

- 小尺度 a →压缩的小波 →快速变换的细节 →高频部分
- 大尺度 a →拉伸的小波 →缓慢变换的粗部 →低频部分

图 1-13 是尺度和频率的示意图。

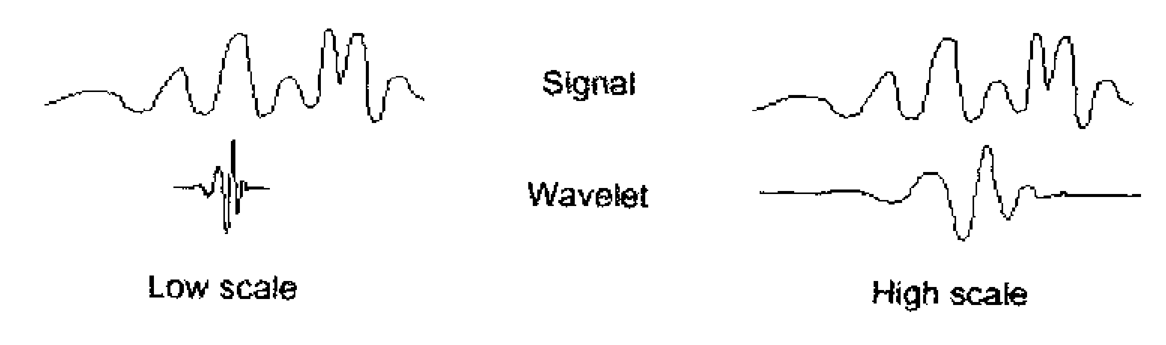


图 1-13 尺度和频率

1.5 离散小波变换

虽然从提取特征的角度看，常常还需要采用连续小波变换，但是在每个可能的尺度离散点都去计算小波系数，那将是个巨大的工程，并且产生一大堆令人讨厌的数据。如果只取这些尺度的一小部分，以及部分时间点，将会大大减轻我们的工作量，同时并不失准确性。离散小波变换主要就是建立在二进制小波变换的基础上的。

目前最通行的办法是对尺度按幂级数进行离散化。即令尺度$a=a_0^0$，a_0^1，a_0^2，…，a_0^j，j=1，2，…，N。当尺度扩大a_0^j倍时，意味着频率降低a_0^j倍，因此采样间隔可以扩大a_0^j倍。一个很自然的想法是将时间位移也以a_0^j倍进行离散化，即沿时间轴以a_0^j为间隔做均匀采样，根据Nyquist采样定理，这样仍然可以不丢失信息。以幂级数进行离散化是一个高效的离散方法，因为幂指数j的小变化，就会引起尺度非常大的变化，动态范围非常大。一般情况下，a取2，这样非常便于分析，并且适合于在计算机上进行高效的运算。

1. 一阶滤波：近似与细节

对于大多数信号来说，低频部分往往是最重要的，往往给出了信号的特征。而高频部分则与噪音及扰动联系在一起。将信号的高频部分去掉，信号的基本特征仍然可以保留。正因为这个原因，我们在信号的分析中，经常会提到对信号的近似与细节。近似主要是系统大的、低频的成分，而细节往往是信号局部、高频成分。信号的滤波过程可以用图1-14表示。

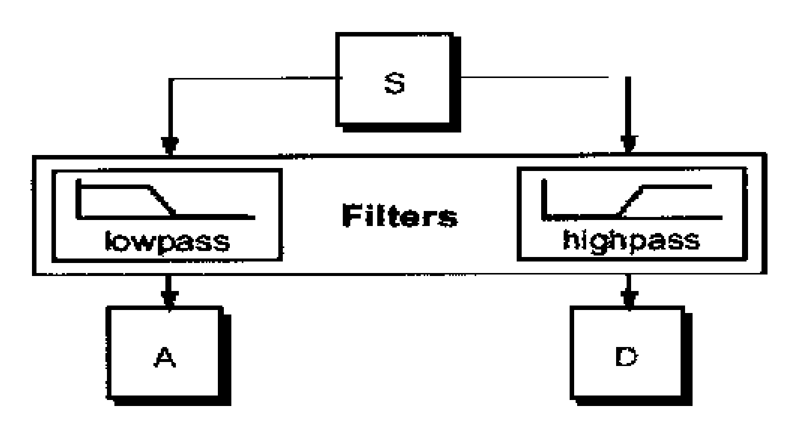


图 1-14　信号的滤波过程

原始信号 S 通过两个互补对称滤波器后，得到两路信号。但如果完全按照滤波器运算进行下去，则得到的数据尺度将是原来数据长度的两倍，这显然存在信息冗余。例如，如果原始信号 S 有 1000 个点，则通过低通滤波器后有 1000 个点，通过高通滤波器后有 1000 个点，得到的信号总长度为 2000 个点，如图 1-15 所示。

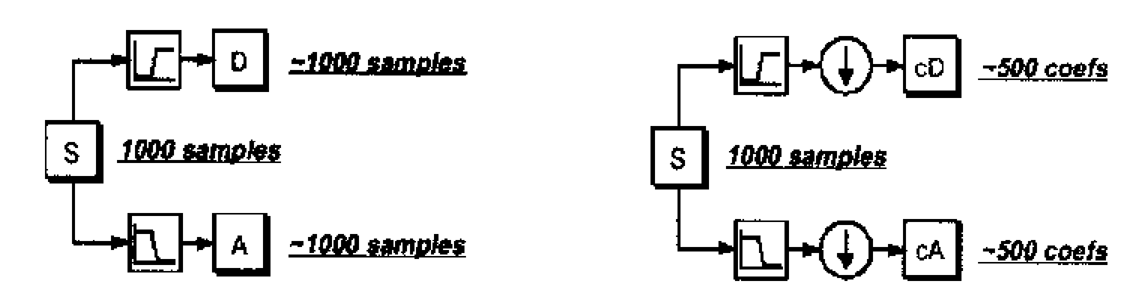


图 1-15　滤波

为了解决这个问题，可以用抽取的方法。即滤波后，每隔一个数据就扔掉一个数据，如图 1-16 所示。这样，数据 S 经过图 1-15 所示的滤波与抽取后，得到了在 MATLAB 小波工具箱中的离散小波变换系数。

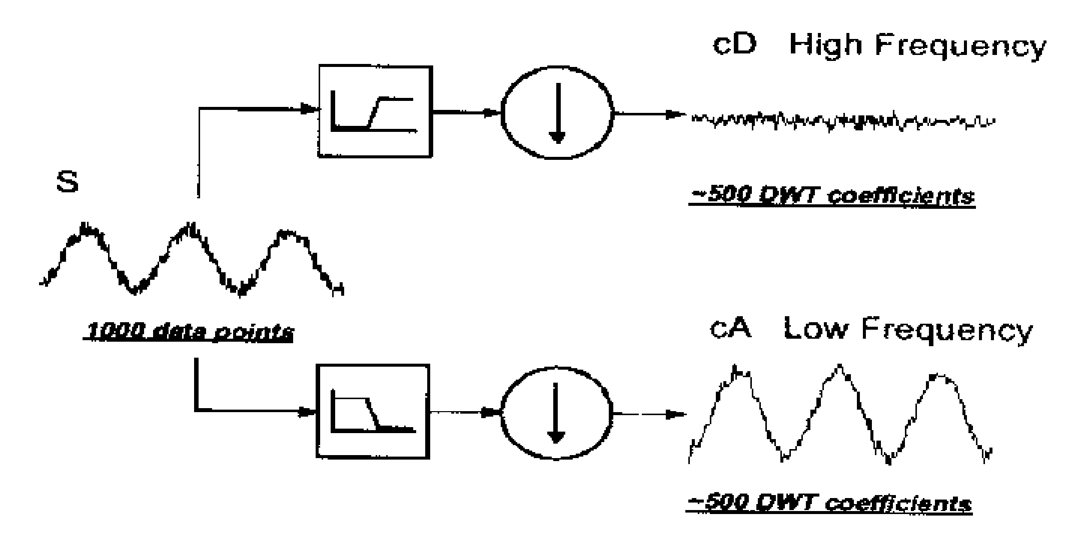


图 1-16　抽取

下面我们来研究一个被噪音污染的正弦信号通过这两个滤波器后的情况。在 MATLB

命令行下，键入下面的命令，就可以得到经过两个互补滤波器后的离散小波变换系数：

```
s = sin(20.*linspace(0,pi,1000)) + 0.5.*rand(1,1000);
[cA,cD] = dwt(s,'db2');
```

2．多尺度分解

上述分解过程可以反复进行，信号的低频部分还可以被继续分解，这样就得到图1-17所示的小波分解树。

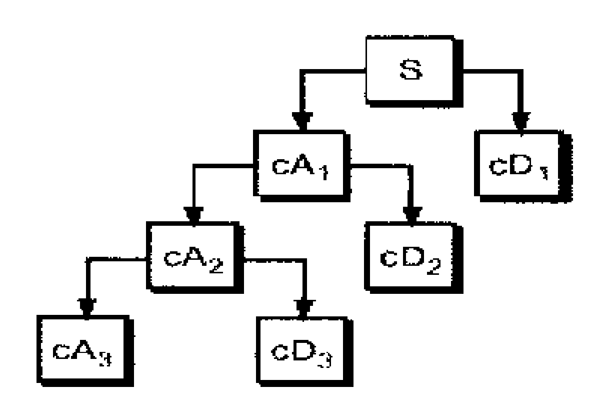


图 1-17　小波分解树

图 1-18 是一个实际的信号进行小波分解的情况。当然，信号分解的层数不是任意的，例如长度为 N 的信号最多能分成 $\log_2 N$ 层。实际中，可以选择合适的分解层数。

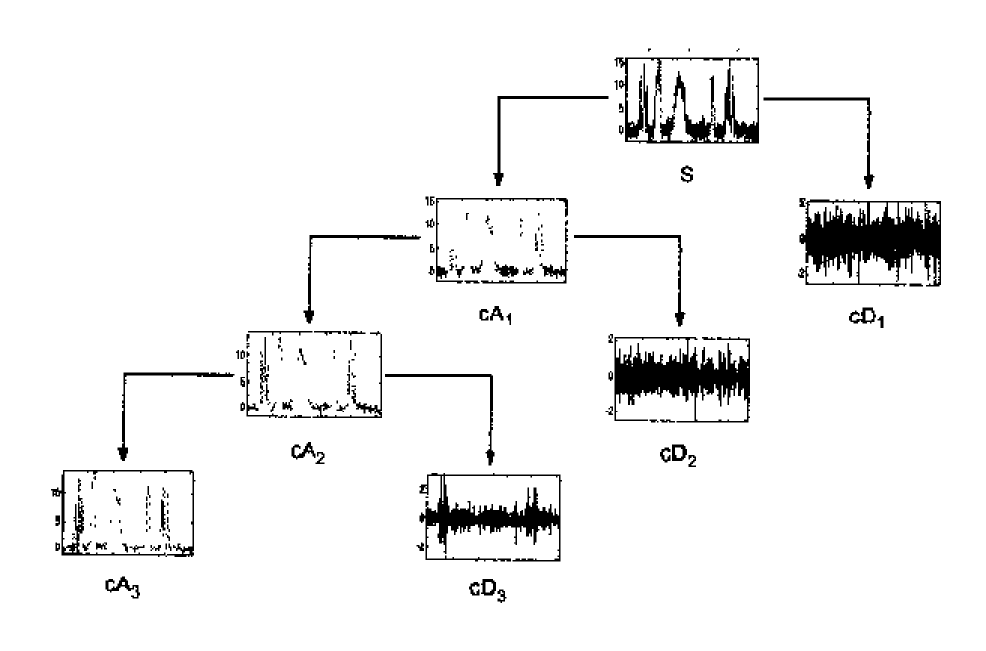


图 1-18　小波分解

3．小波重构

将信号分解成一个个互相正交小波函数的线性组合，可以展示信号的重要特性，但这并不是小波分析的全部。小波分析另一个重要的方面就是在分析、比较、处理（如去掉高频信号、加密等）小波变换系数后，根据新得到系数去重构信号。这个过程称之为逆离散小波变换（IDWT），或小波重构、合成等。信号重构的基本过程如图 1-19 所示。

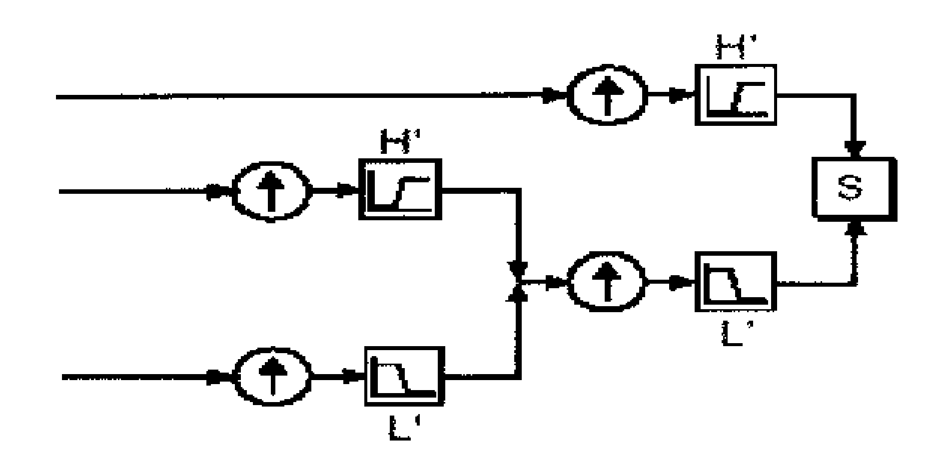


图 1-19　信号重构的基本过程

信号合成主要包括对小波变换系数的插值与滤波，正好与信号分解相反。

4．重构滤波器

重构滤波器的类型对信号重构的质量非常关键，值得我们认真讨论一下。事实证明，选择合适的分解与重构滤波器对于信号分析的质量非常关键。在信号分解与综合的过程中，由于存在抽取与插值，从而有信号频率混叠的可能性。必须选择合适的滤波器将这种混叠消除掉。低通分解滤波器（Lo_D）、高通分解滤波器（Hi_D），以及相应的低通重构滤波器（Lo_R）、高通重构滤波器（Hi_R），和起来构成镜像滤波器组，如图 1-20 所示。

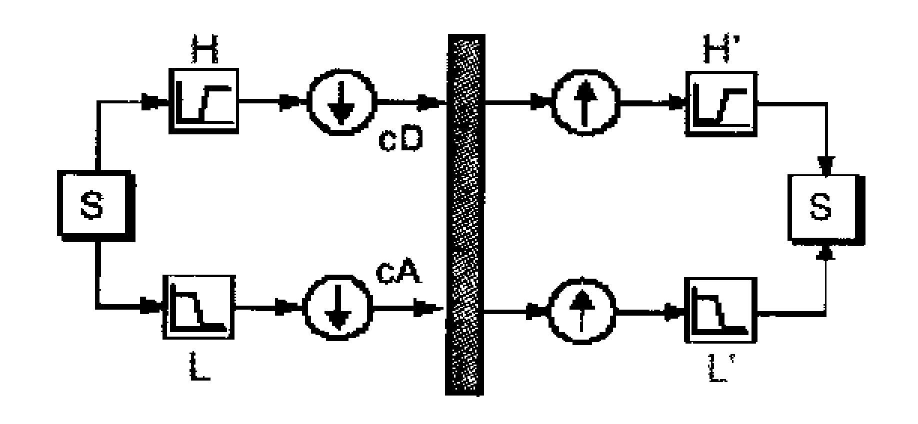


图 1-20　镜像滤波器组

5．重构粗略部分与细节部分

可以通过对信号经过分解滤波器后得到的系数进行重构，对信号进行重构。考虑上例得到的500个低通滤波器系数cA与高通系数cD，经过插值与滤波后得到原始信号，如图1-21所示。但也可以只对cA或cD进行重构，相当于对信号的粗略部分与细节部分进行重构，如图1-22和图1-23所示。

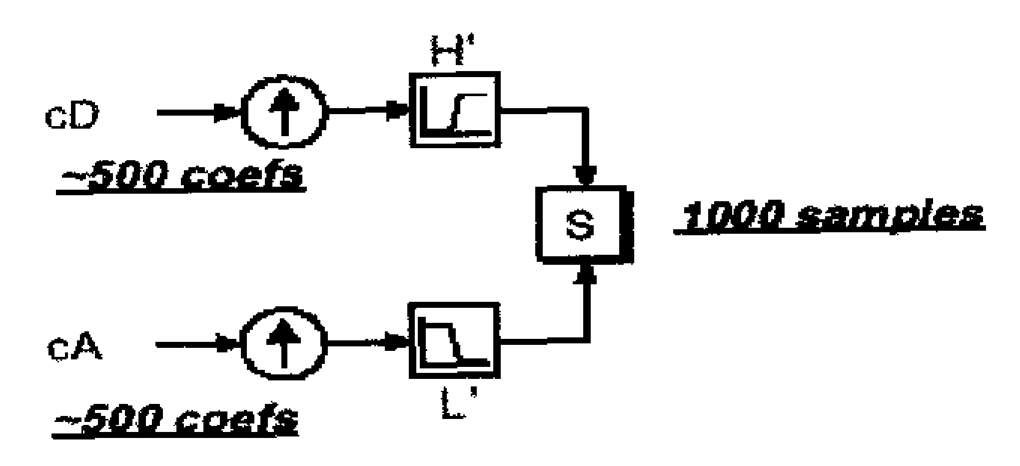


图 1-21　原始信号

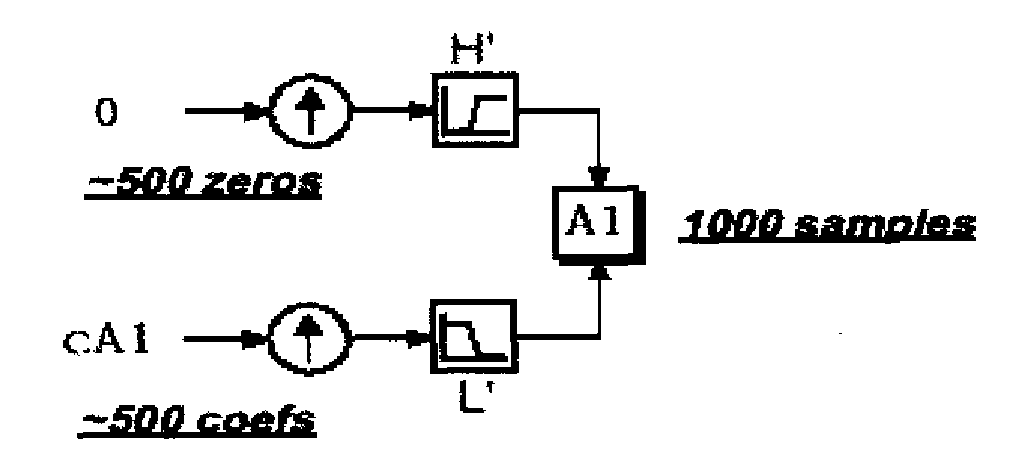


图 1-22　重构粗略部分

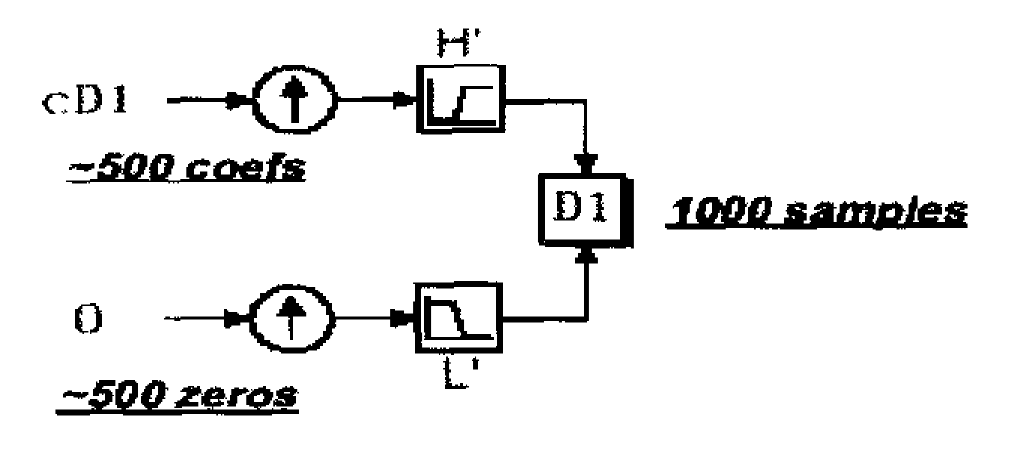


图 1-23　重构细节部分

6．多尺度分解与重构

多尺度分解与重构基本与以上分解与重构一致，包括将信号分解成多层小波系数，然后对这些系数进行分析、处理（如让其中一些系数为 0、对部分系数加密等），对得到的新系数进行多尺度重构。对系数进行处理的过程，其作用相当于滤波、消噪、压缩、加密等过程，灵活性相当大。如图 1-24 所示。

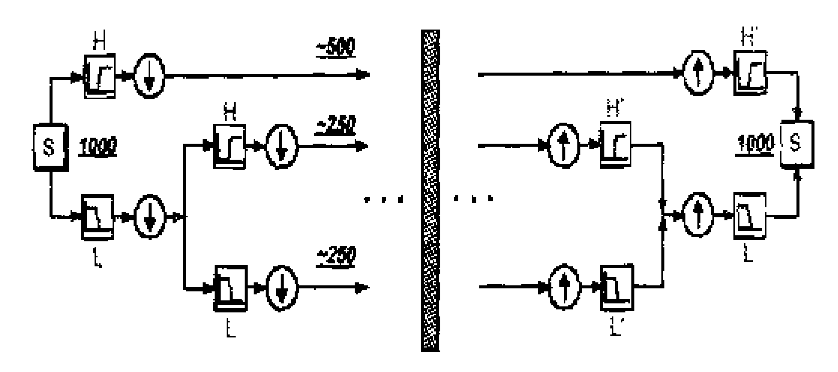


图 1-24 多尺度分解与重构

1.6 小波包分析

小波包分析是从小波分析延伸出来的一种对信号进行更加细致的分析与重构的方法在以上的分析中，大家已经知道，在小波分析中，实际上是将信号分解成低频的粗略部分与高频的细节部分，然后只对低频细节再做第二次分解，分解成低频部分与高频部分，而不对高频部分做第二次分解，如图 1-25 所示。依次类推得到的分解系数序列即为小波分解的系数结果。

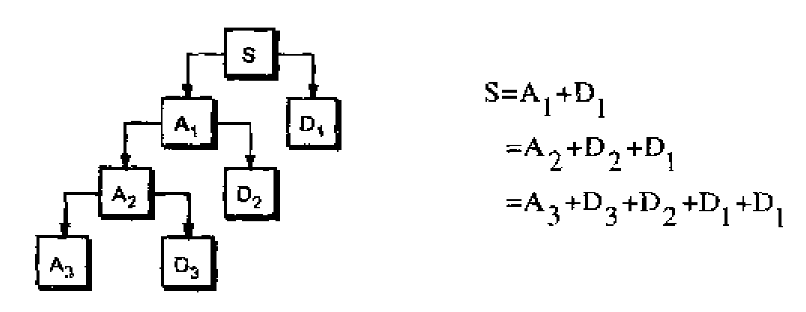


图 1-25 小波包分析——低频分解

小波包分析不但对低频部分进行分解，而且对高频部分也做了二次分解，如图 1-26 所示。信号 S 可以被表示成 $S=A_1+AAD_3+DAD_3+DD_2$。

小波包的主要优点是小波包可以对信号的高频部分做更加细致的刻画，对信号的分析能力更强，当然其代价是信号分析的计算量将显著上升。

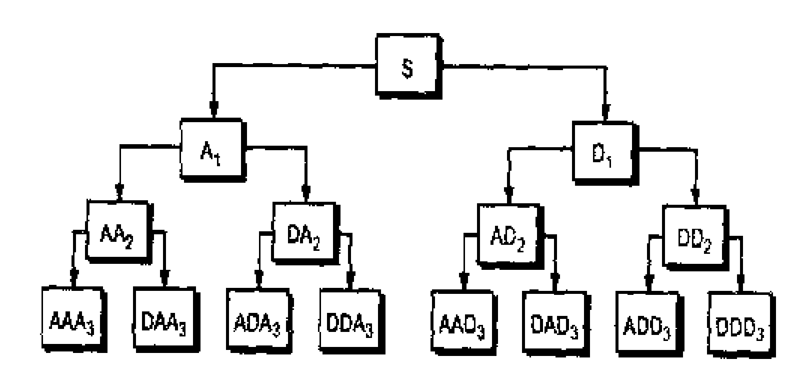


图 1-26 小波包分析——高频分解

1.7 常用小波函数

并不是随便一个函数都可以成为小波基函数，要满足非常严格的限制方可以成为一个可用的小波函数，进而发展成一个好的小波变换函数。因此实用的小波将非常难找、一些小波函数甚至没有解析表达式。许多小波分析的专著都有详细讨论，本书不做过多评述。本书主要介绍 MATLAB 6.5 小波工具箱中使用的小波。

使用 wavemngr('read',1)命令可以显示 MATLAB 6.5 小波工具箱中的所有小波：

```
>> wavemngr('read',1)

ans =

===================================
Haar                    haar
===================================
Daubechies              db
------------------------------
db1  db2  db3  db4
db5  db6  db7  db8
db9  db10 db**
===================================
Symlets                 sym
------------------------------
sym2 sym3 sym4 sym5
sym6 sym7 sym8 sym**
===================================
Coiflets                coif
------------------------------
coif1 coif2 coif3 coif4
coif5
===================================
BiorSplines             bior
------------------------------
bior1.1    bior1.3    bior1.5    bior2.2
bior2.4    bior2.6    bior2.8    bior3.1
bior3.3    bior3.5    bior3.7    bior3.9
bior4.4    bior5.5    bior6.8
===================================
ReverseBior             rbio
------------------------------
rbio1.1    rbio1.3    rbio1.5    rbio2.2
rbio2.4    rbio2.6    rbio2.8    rbio3.1
rbio3.3    rbio3.5    rbio3.7    rbio3.9
rbio4.4    rbio5.5    rbio6.8
```

```
==================================
Meyer                    meyr
==================================
DMeyer                   dmey
==================================
Gaussian                 gaus
------------------------------
gaus1 gaus2gaus3gaus4
gaus5gaus6gaus7gaus8
gaus**
==================================
Mexican_hat              mexh
==================================
Morlet                   morl
==================================
Complex Gaussian         cgau
------------------------------
cgau1    cgau2    cgau3    cgau4
cgau5    cgau**
==================================
Shannon                  shan
------------------------------
shan1-1.5  shan1-1   shan1-0.5  shan1-0.1
shan2-3    shan**
==================================
Frequency B-Spline       fbsp
------------------------------
fbsp1-1-1.5fbsp1-1-1  fbsp1-1-0.5fbsp2-1-1
fbsp2-1-0.5fbsp2-1-0.1fbsp**
==================================
Complex Morlet           cmor
------------------------------
cmor1-1.5  cmor1-1   cmor1-0.5  cmor1-1
cmor1-0.5  cmor1-0.1  cmor**
==================================
Lemarie                  lem
------------------------------
lem1 lem2 lem3 lem4
lem5
==================================
```

当然读者也可以将新出现的小波加入到小波工具箱中。

1．Haar

Haar 小波是所有小波中最简单的，有一个有限的紧支撑，计算比较简单。Haar 函数的定义是：

$$\psi = \begin{cases} 1 & 0 \leqslant x \leqslant 1/2 \\ -1 & 1/2 \leqslant x < 1 \\ 0 & \text{其他} \end{cases}$$

其小波函数与尺度函数如图 1-27 所示。

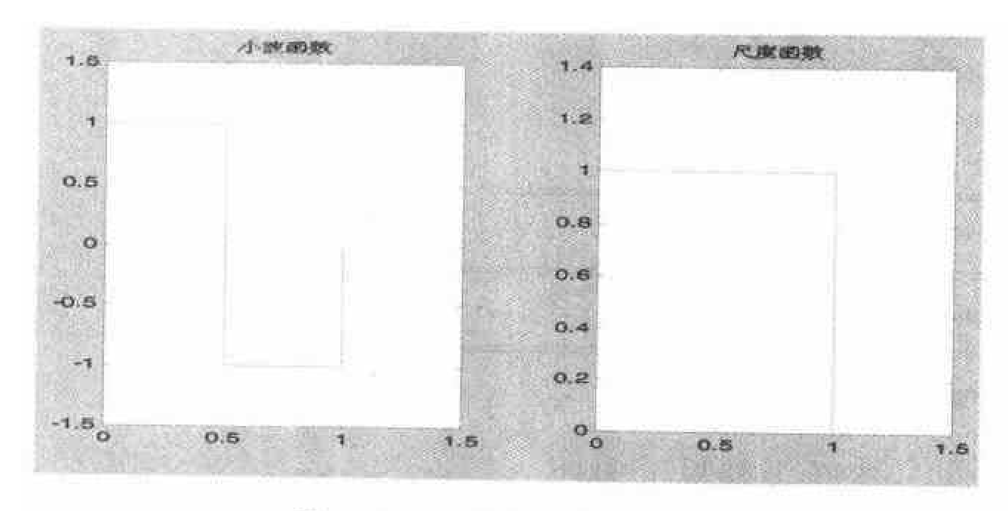


图 1-27　Haar 小波函数与尺度函数

2．Daubechies

Daubechies 小波系是由法国学者 Daubechies 提出的一系列二进制小波的总称。在 MATLAB 中记为 db$N$，N 为小波的序号，N 值取 2，3，…，10。该小波没有明确的解析表达式，小波函数ψ 与尺度函数ϕ 的有效支撑长度为 $2N-1$，小波函数ψ 的消失矩为 N。

图 1-28 给出了 N=3、5、7、9、10 时 Daubechies 小波的尺度函数与小波函数图形。

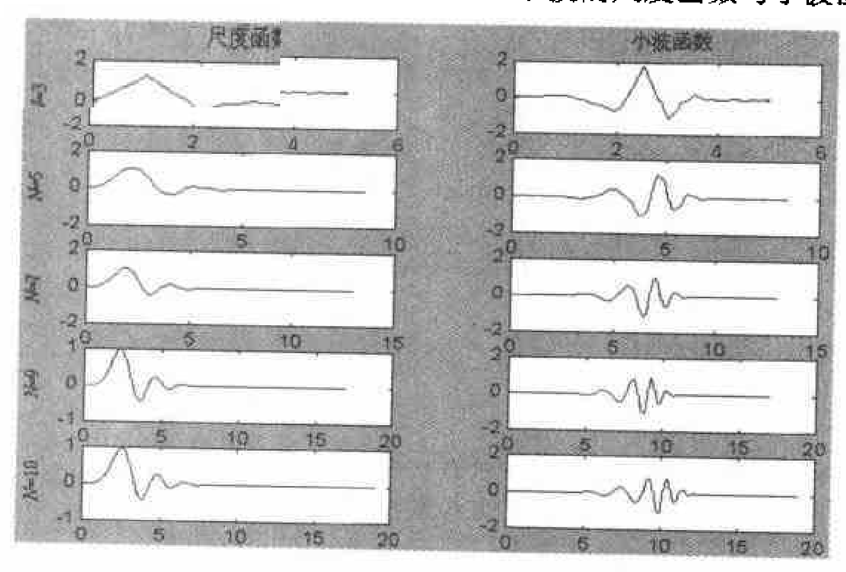


图 1-28　Daubechies 小波的小波函数与尺度函数

3．Symlets

Symlets 小波系也是一系列小波的总称，其编号为 sym$N$，N=2，3，4，…。这类双正

交小波的支撑长度为 $2N-1$，滤波器的长度为 $2N$，消失矩为 N，具有近似的对称性。图 1-29 为 N=2、3、6、9 时小波的尺度函数与小波函数图形。

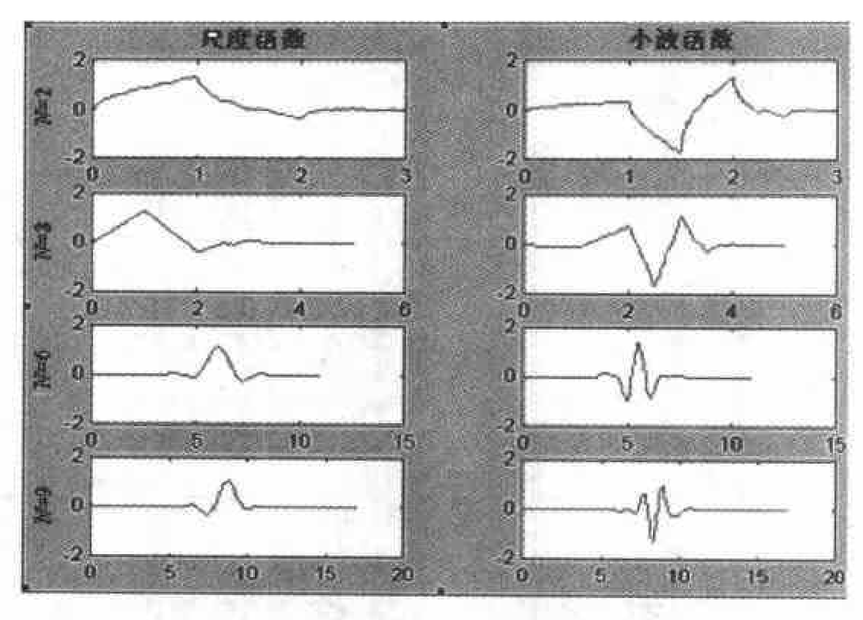


图 1-29　Symlet 小波的尺度函数与小波函数

4．Coiflets

Coiflet 小波系具有 coif$N$（N=1，2，3，4，5）五种小波，为双正交的小波，其支撑长度为 $6N-1$，滤波器的长度为 $6N$，消失矩为 $2N$。该小波比 db$N$ 小波对称性要好。图 1-30 为各阶小波的小波函数。

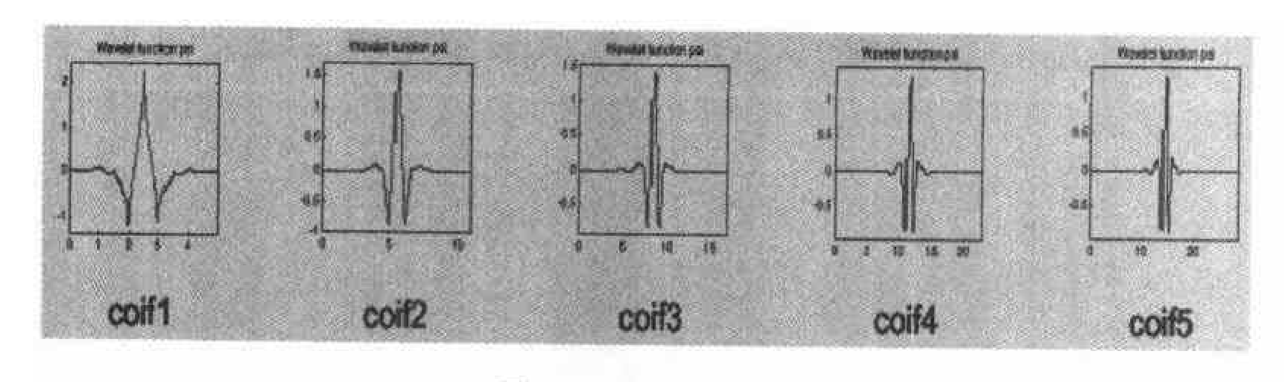


图 1-30　Coiflet 小波系

5．Biorthogonal

该小波系主要特点是具有线性相位，可以应用于信号与图像的重构中，通常表示成 bio$N$r.Nd 的形式：

Nr=1　　Nd=1，3，5

Nr=2　　Nd=2，4，6，8

Nr=3　　Nd=1，3，5，7，9

Nr=4　　Nd=4

Nr=5　　Nd=5

Nr=6　　Nd=8

d 表示分解，r 表示重构。图 1-31 给出了这种小波系的小波函数。

图 1-31　Biorthogonal 小波系

6．Meyer

Meyer 小波的小波函数与尺度函数如图 1-32 所示。

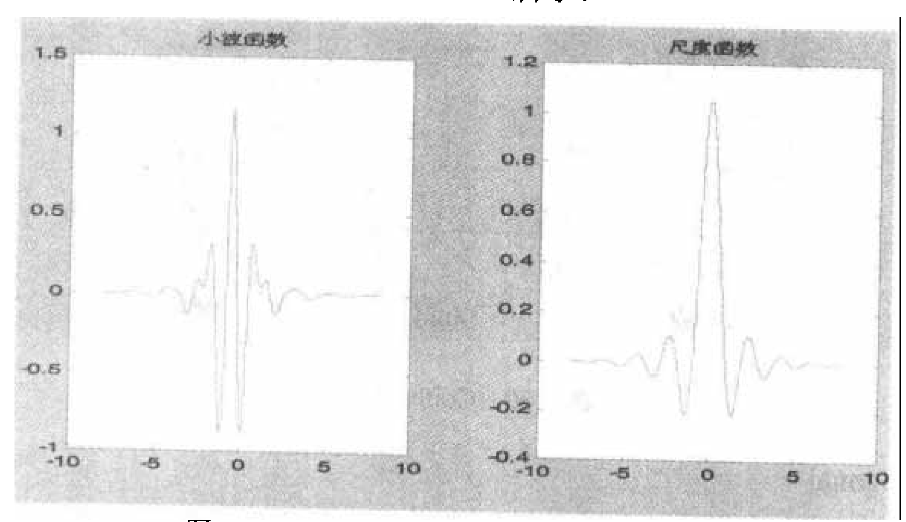


图 1-32　Meyer 小波的小波函数与尺度函数

7．Dmeyer

Dmeyer 即离散的 Meyer 小波。其小波函数与尺度函数如图 1-33 所示。

8．Gaussian

Gaussian 小波是高斯密度函数的微分形式，是一种非正交与双正交的小波，没有尺度函数，其小波函数如图 1-34 所示。

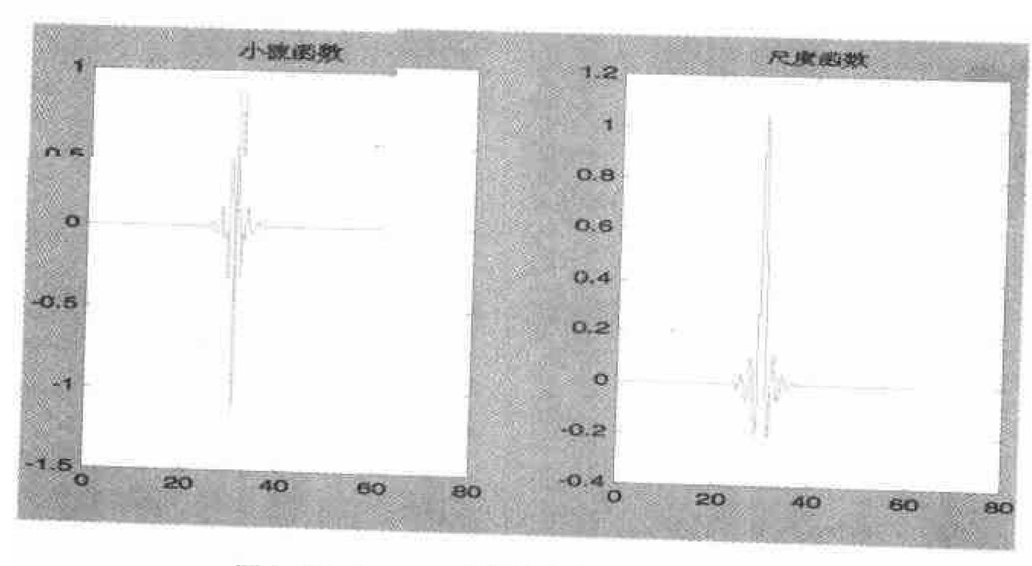


图 1-33　Dmeyer 小波的小波函数与尺度函数

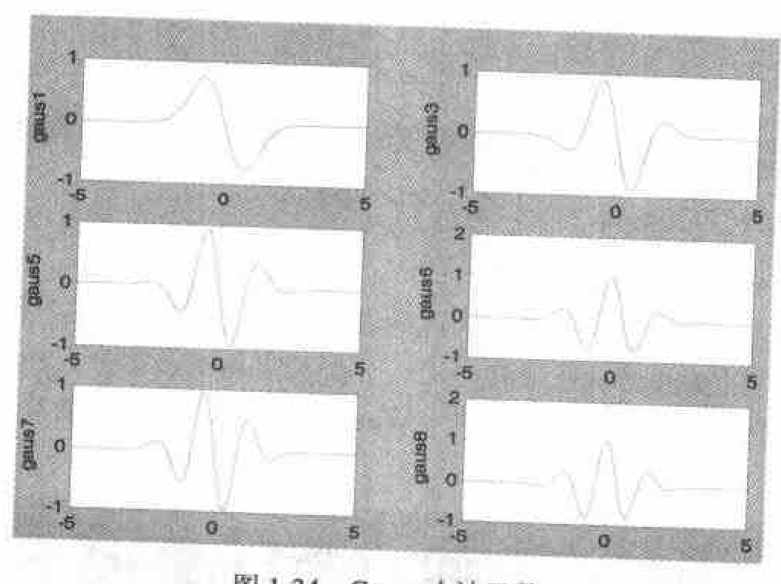


图 1-34　Gauss 小波函数

9. Mexican hat

Mexican hat 小波函数是 Gauss 函数的二阶导数，其小波函数的图像有点像墨西哥帽，故得此名。其尺度函数不存在。图 1-35 是其小波函数波形。

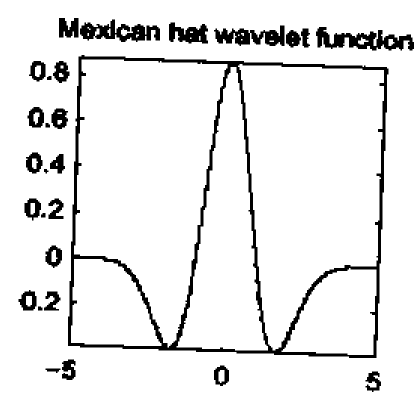


图 1-35　Mexican hat 小波函数

10. Morlet

Morlet 小波的定义为：

$$\psi(x)=Ce^{-\frac{x^2}{2}}\cos 5x$$

其小波函数的图像如图 1-36 所示。

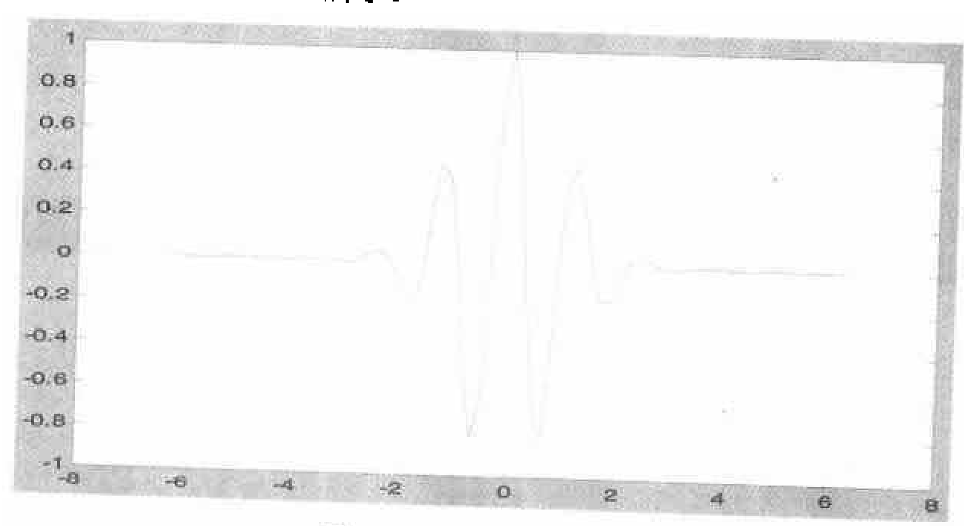


图 1-36 Morlet 小波函数

11. ReverseBior

ReverseBior 由 biorthogonal 而来，因此两者形式很类似，图 1-37 是 rbio1.5 的尺度函数、小波函数及相应的滤波器（低通和高通、分解和重构滤波器）。用户可以在小波工具箱的图形接口界面中的小波信息显示工具（Wavelet Display）中查看相应小波函数的更详细的信息。

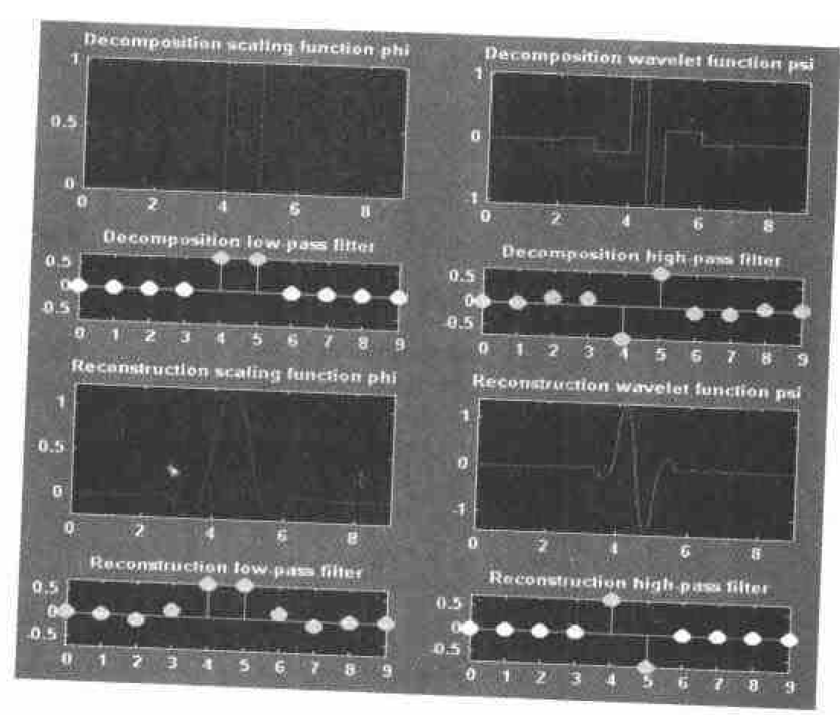


图 1-37 ReverseBior 小波函数

12. Complex Gaussian

Complex Gaussian 属于一类复小波，图 1-38 是 cgau8 小波函数的实部和虚部、模和相位角。

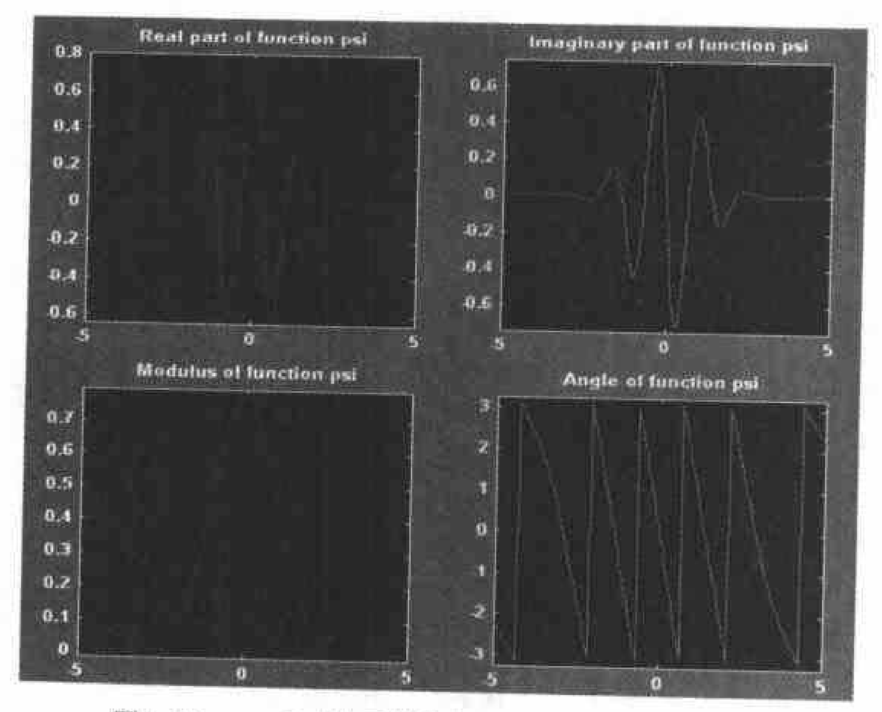


图 1-38 cgau8 小波函数的实部和虚部、模和相位角

13. Complex Morlet

Complex Morlet 也是一类复小波，图 1-39 是 cmor1.5 的小波函数的实部和虚部、模和相位角。

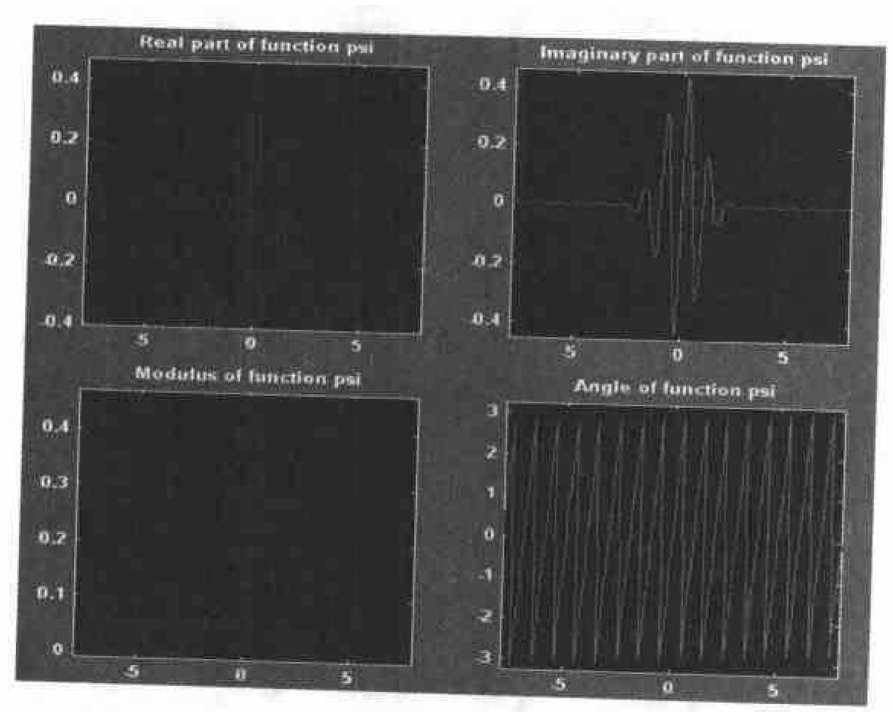


图 1-39 cmor1.5 的小波函数的实部和虚部、模和相位角

14. Complex Frequency B-Spline Wavelets

图 1-40 是 fbsp1-1-1.5 小波函数的实部和虚部、模和相位角。

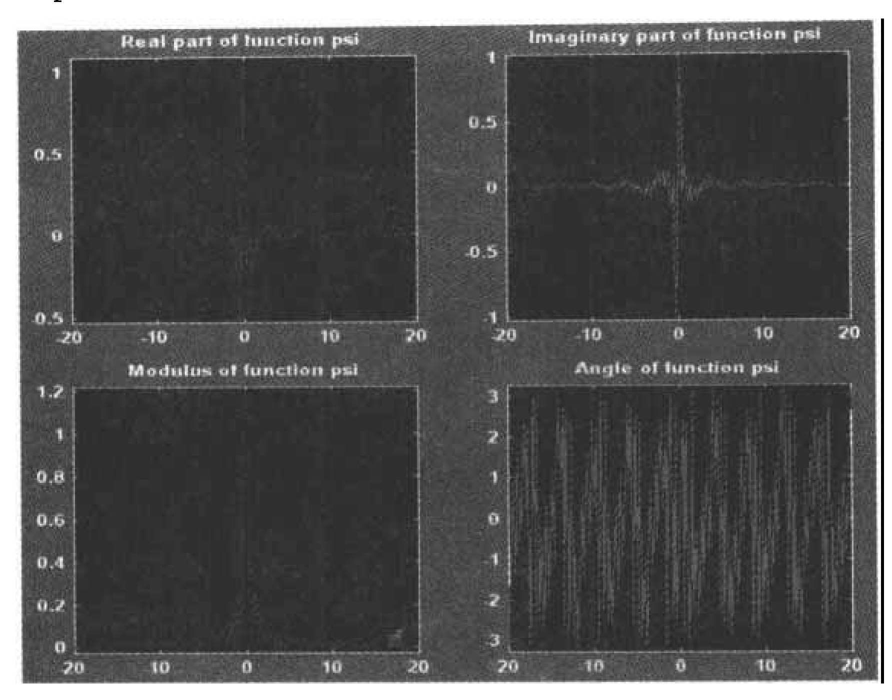


图 1-40 fbsp1-1-1.5 小波函数的实部和虚部、模和相位角

15. Complex Shannon Wavelets: shan

图 1-41 是 shan1-1.5 小波函数的实部和虚部、模和相位角。

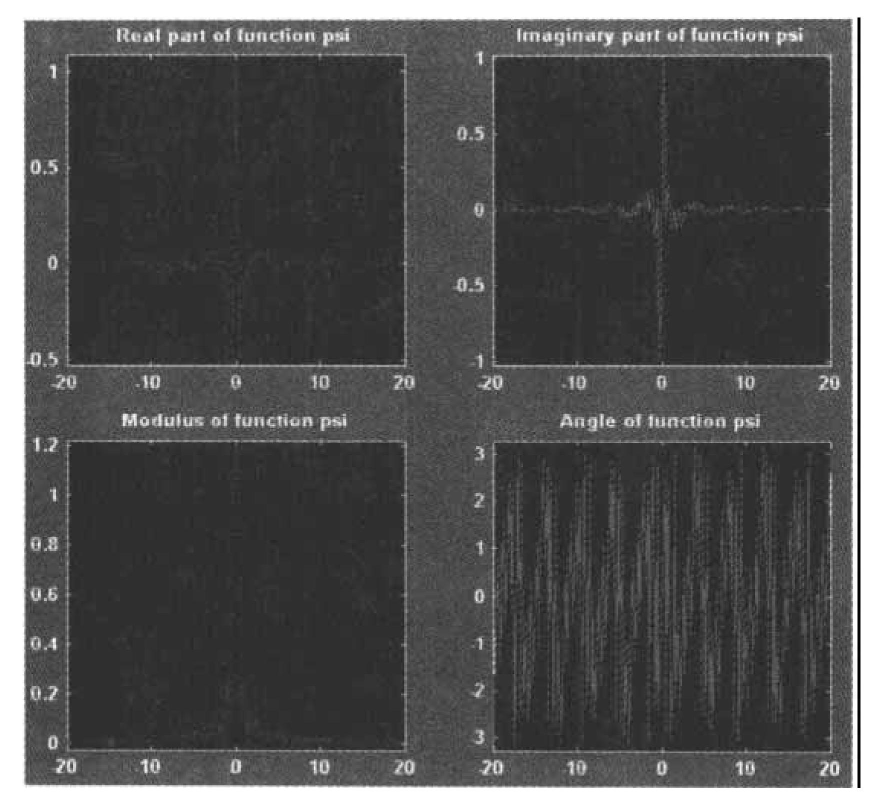


图 1-41 shan1-1.5 小波函数的实部和虚部、模和相位角

第 2 章　MATLAB 6.5 小波分析示例

前面一章介绍了小波分析的基础理论，而 MATLAB 6.5 语言为我们提供了非常好的小波分析实现工具，下面我们将以大量的例子来展示 MATLAB 6.5 环境下进行小波分析的方法和技巧。

本章主要内容：

- 一维连续小波分析
- 一维连续复小波分析
- 一维离散小波分析
- 二维离散小波分析
- 一维离散平稳小波分析
- 二维离散平稳小波分析
- 一维小波回归估计
- 一维小波密度估计
- 一维小波系数的自适应阈值处理
- 小波系数选取
- 一维信号延拓或截断
- 二维信号延拓或截断

2.1　一维连续小波分析

本节介绍如何利用小波工具箱的命令行和图形工具两种方式来进行一维连续小波分析。最后讨论了如何在磁盘和图形接口工具之间交换信号和系数信息。

2.1.1　一维连续小波分析——命令行方式

小波工具箱只需要一个函数来实现连续小波分析：cwt。

其格式为：

①coefs＝cwt（s,scale, 'wname'）

②coefs＝cwt（s,scale, 'wname', 'plot'）

说明：其中 coefs 为连续小波变换后的返回系数 W_f（a,b）矩阵，系数以行方向存储在矩阵中。因为 f(t)是一个离散信号，所以可以用 f(k)的形式表示信号，k 从 1 到 length(f)，所以对于一具体的尺度 a，cwt 将计算 b 从 1 到 length(f)所对应的每一个系数值 W_f（a,b）。

s 为待分析的信号，在计算中通常是以离散的形式给出，如 MATLAB 自身所带有的 nossion 信号。scales 为连续小波变换的尺度向量，wname 为小波函数名。如果尺度为离散

的值 a1,a2,a3,…，则尺度向量 scales 形式为：[a1,a2,a3,…]，值与值之间用逗号或者空格分开；假如尺度向量 scales 由起始尺度 amin、终止尺度 amax、步长 astep 三者组成，则尺度向量 scales 可表示为：[amin：astep：amax] 的形式（中括号“[]”此时可省略）；假如同时包括上面两种情况，则把离散尺度写在前面，步长表示方式写在后面：[a1，a2，a3，amin：astep：amax]。Wname 是小波函数名，如 haar，db1，db2，meyer 等。返回系数矩阵 coefs 的行数为小波变换中尺度的个数，列数为信号采样点的个数，如 nossion 信号的采样点个数为 1000，矩阵第一行的值对应第一个尺度变换后的系数，第二行的值对应第二个尺度变换后的系数，依此类推。对于格式②，“plot”是用来画出小波变换后系数的图形，在图形中，系数的大小是以灰度的深浅来表示，颜色越深，则变换后的系数 $W_f(a,b)$ 越大。



（1）尺度必须为正实数；（2）逗号可以换成空格的形式；（3）当尺度步长 astep 没有显式给出时，则表示 astep 取默认值 1。

如图 2-1 所示是以一个含有噪声的正弦信号为例，说明如何利用小波工具箱函数进行小波分析。

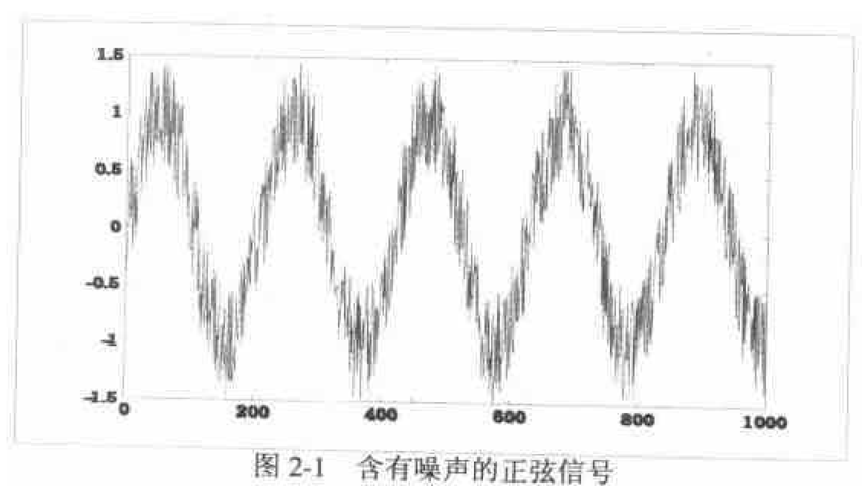


图 2-1　含有噪声的正弦信号

1．装载信号

在 MATLAB 命令行窗口中输入：

```
>>load noissin;
```

则通过下面的命令可查看工作区的“noissin”信号：

```
>>whos;
  Name        Size        Bytes      Class
  noissin     1×1000      8000       double array
```

2．完成一维连续小波变换

```
>>c=cwt(noissin,1:48,'db4');
```

说明：其中 c 为一个 48×1000 的矩阵，每行对应相应某个尺度变换后的系数。

3．产生系数的图形表示

产生系数的图形表示如图 2-2 所示（彩色效果图见彩插 1）。

```
>>c=cwt(noissin,1:48,'db4','plot');
```

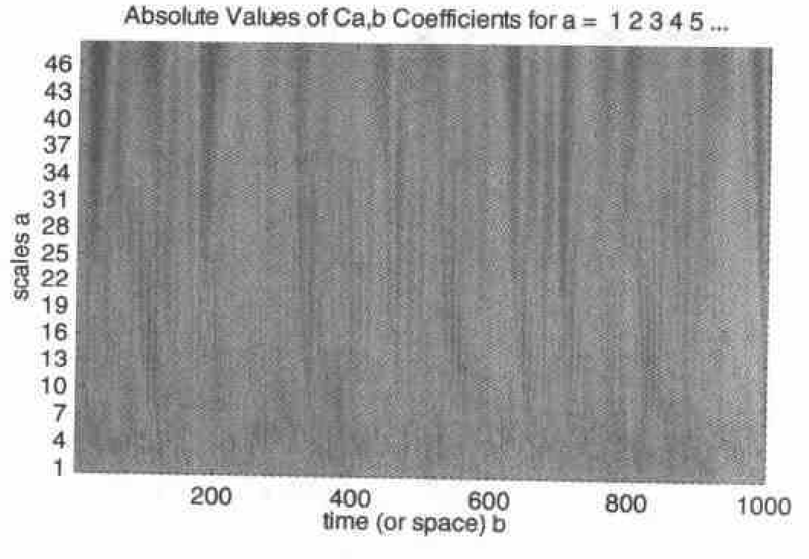


图 2-2　系数的图形表示

4. 重新选择分析尺度

```
>>c=cwt(noissin,2:2:128,'db4','plot');
```

结果如图 2-3 所示。可以看出，在新的尺度下正弦信号的周期性较明显。

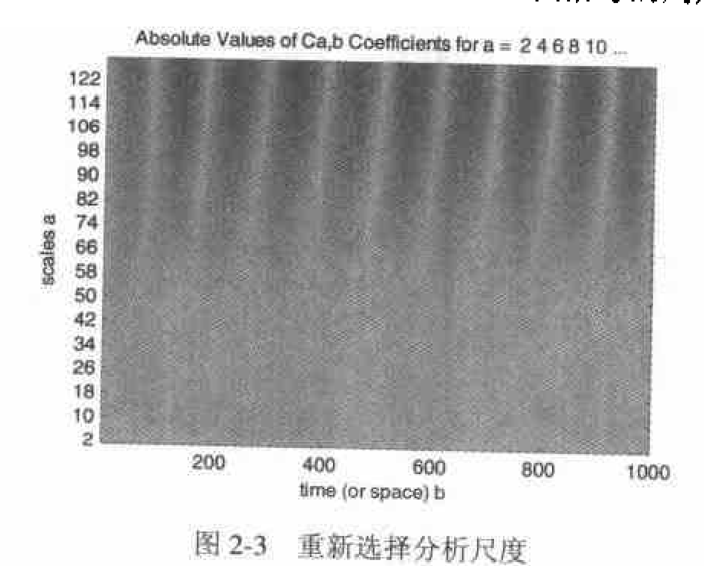


图 2-3　重新选择分析尺度

2.1.2　一维连续小波分析——图形接口方式

现在我们采用图形接口方式中的 Continuous Wavelet 1-D 工具来完成上面的分析过程。

1. 启动 Continuous Wavelet 1-D 工具

在 MATLAB 命令符下键入 wavemenu，出现小波工具箱主菜单窗口（Wavelet Toolbox Main Menu），如图 2-4 所示。选择 Continuous Wavelet 1-D，出现如图 2-5 所示的一维连续小波图形工具。

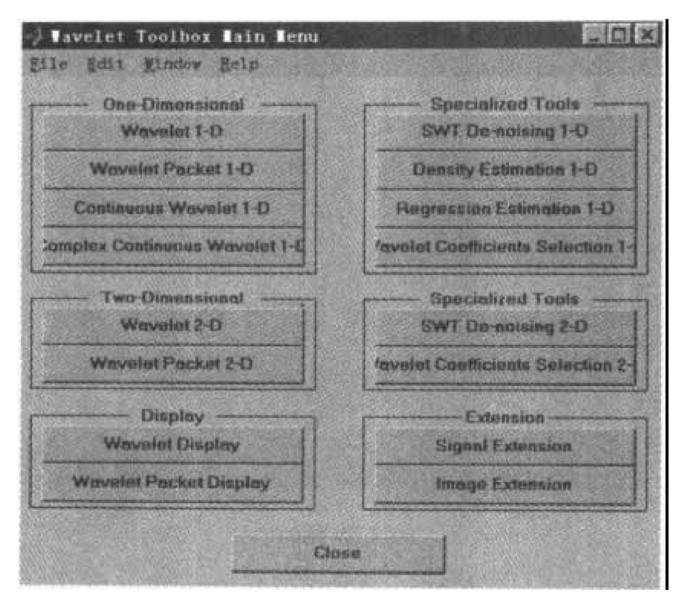


图 2-4 小波工具箱主菜单窗口

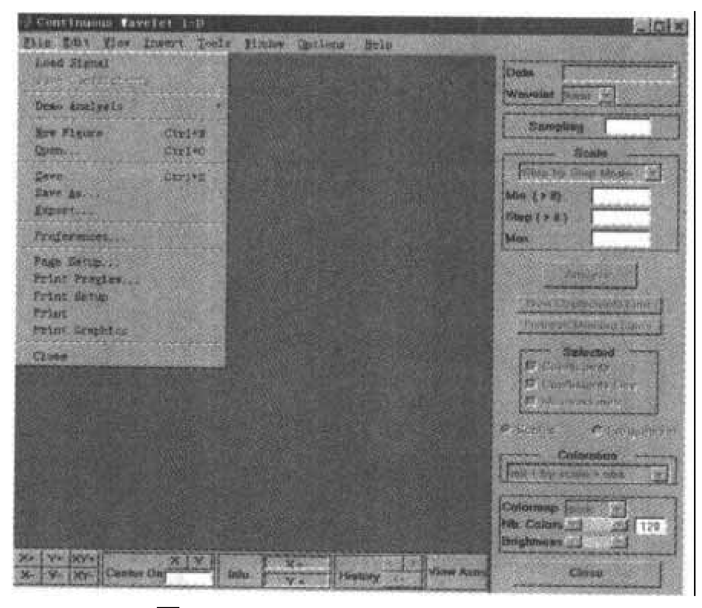


图 2-5 一维连续小波图形工具

2. 装载信号

单击图 2-5 中的【File】→【Load Signal】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noissin.mat 文件。

3. 完成一维连续小波变换

在图 2-5 中的右上角选择基本小波为 db4 和尺度为 1：1：48。注意，在小波工具箱提供的图形用户界面 GUI（Graphical User Interface）中，小波函数有 11 种，即 haar，db，bior，sym，coif，rbio，meyr，dmey，gaus，mexh 和 morl。染色模式（Coloration Mode）选择 init＋by scale。选择 Init 时是对所有的系数都进行染色，即将这些数据分成若干等份（等份的数量与右下方的颜色数目 Nb.Colors 相同），然后将所有的数据以对应的颜色显示出来。选择 by scale（只在一维连续小波分析中适用），表示单独对每一个高频层或高频尺度进行染色；否则，对所有分解层或分解尺度都染色。窗口右下角还有一个 Colormap 选择框，用



于调节颜色映射，使二维图像的显示或一维曲线的绘制以相应的颜色映射显示出来。

选择好以上参数后，就可以单击【Analyze】按钮。经过短时间的计算后将出现如图 2-6 所示的分析结果。

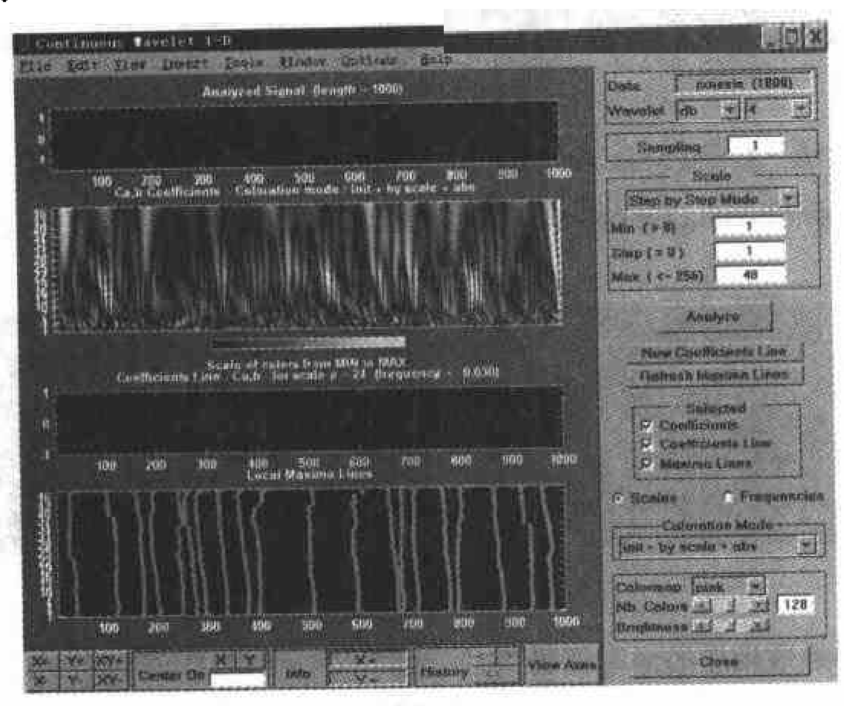


图 2-6　分析结果

4．观察小波系数线（Wavelet Coefficients Line）

将鼠标移到图 2-6 中系数图形（第二个图形）上按住右键不放，则在底部的 Info 框中将指示出鼠标所在位置对应的位置信息（X）和尺度信息（Sca），如图 2-7 所示。移动鼠标直到指示信息 Sca 为 40，使尺度 a=40。单击【New Coefficients Line】按钮，则小波系数图（Coefficients Lines）相应更新，如图 2-8 所示。

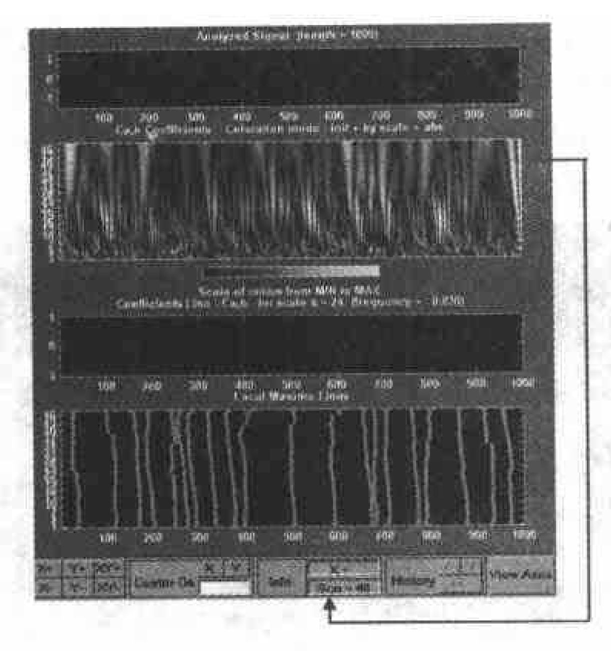


图 2-7　位置信息（X）和尺度信息（Sca）

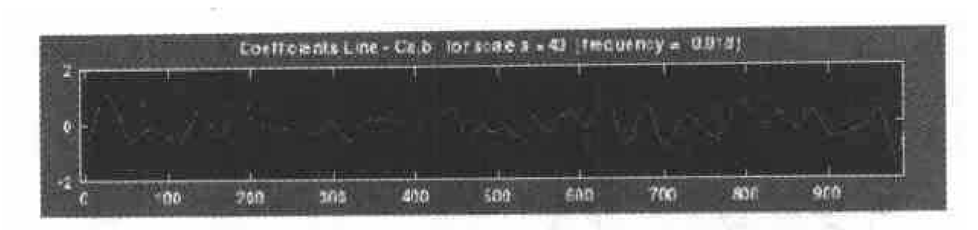


图 2-8 小波系数线

5．观察最大值（Local Maxima Line）

单击【Refresh Maxima Lines】按钮，给出尺度 a 从 40 到 1 对应的小波系数的最大值，如图 2-9 所示。

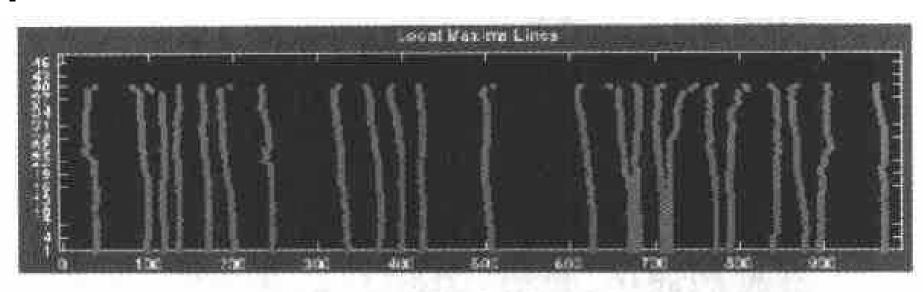


图 2-9 小波系数的最大值

6．将尺度转换成相应的频率信息

在图 2-6 的右中部选择 Frequencies 单选按钮，然后在系数图上按住鼠标右键不放，同选择 Scales 一样，在底部的 Info 框中将指示出鼠标所在位置对应的位置信息（X）和频率信息（Frq，单位 Hz）。小波工具箱的这种在尺度和频率信息之间切换的功能可以让用户了解尺度和频率的对应关系，这种对应关系依赖于小波基函数和采样周期。

同时，用户还可以通过图 2-6 右边的 Selected 选择框来关闭或打开显示相应的图形，以利于观察。

7．细节放大

按住鼠标左键不放，在感兴趣的信号部分拖放一个矩形框。然后单击图 2-5 底部的【X+】按钮，即只在水平方向上放大。则出现图 2-10 所示的放大了的信号和对应的系数图。

就像在前面命令行方式中一样，我们也可以很方便地改变分析尺度或小波基函数重复上面的分析工程，只需要在相应的设置框中重新设置即可。

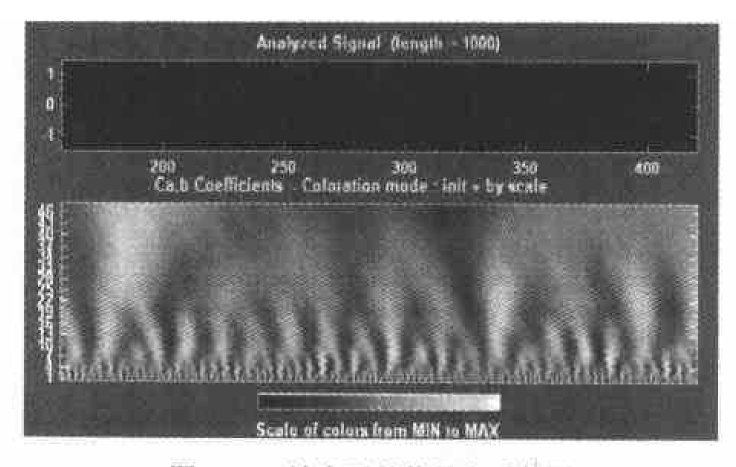


图 2-10 放大了的信号和系数图



8. 系数的绝对模式或正常模式显示

在图 2-6 右下方的染色模式（Coloration Mode）选择框中如果选择 abs，则在系数染色前，先将系数取绝对值，否则系数按正常模式显示。两种显示模式分别如图 2-11 和图 2-12 所示。

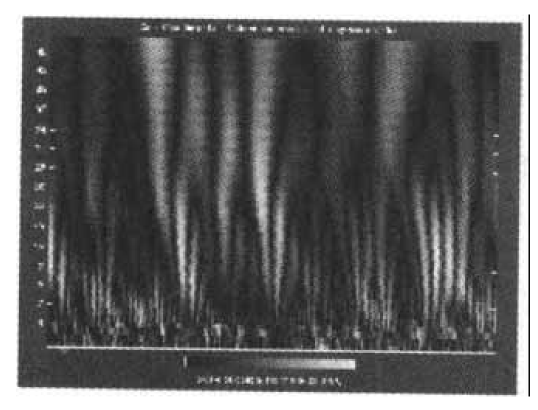


图 2-11　绝对模式

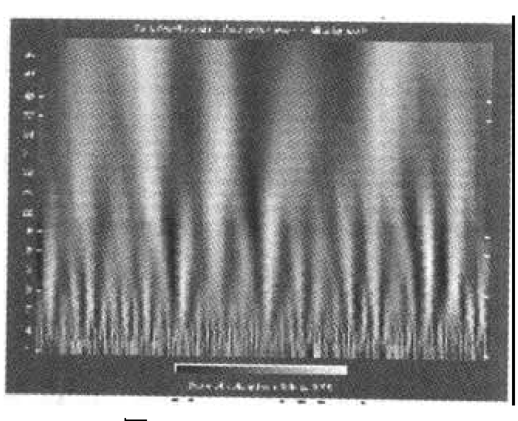


图 2-12　正常模式

2.1.3　图形接口方式中的信息交互

一维连续小波图形工具可以让用户从磁盘引入或输出信息。

1. 从磁盘装载自己定义的信号到一维连续小波图形工具

为了装载用户在 MATLAB 工作区构造的信号到一维连续小波图形工具中，必须将其保存为 MAT 格式的文件。例如：假设用户自己定义了一个名为 warma 的信号并想利用一维连续小波图形工具来进行分析，那么首先键入下述命令将其保存以 MAT 格式在工作区：

```
>>save warma
```

工作区变量必须是一个向量：

```
>>sizewarma=size(warma)
```

然后就可以在一维连续小波图形工具中通过【File】→【Load Signal】菜单命令来装载该信号了。

2. 保存一维连续小波图形工具产生的小波系数到磁盘

在一维连续小波图形工具中单击【File】→【Save Coefficients】菜单命令，则会出现一个保存小波系数的对话框。该对话框在指定的目录下以用户指定的名称产生扩展名为.wc1 的一个 MAT 格式的文件。

例如，一维连续小波图形工具中的一个演示例子：

单击【File】→【Demo Analysis】→【with at scales〔1：1：64)】→【Cantor curve】菜单命令。

当以 cantor.wc1 保存好小波系数后，将其读取到工作区：

```
>>load cantor.wc1 –mat
>>whos
  Name          Size          Bytes          Class
  ans           1×1           8              double array
```

```
coefs        64×2188      1120256      double array
scales       1×64         512          double array
```

说明：可以看出，变量 coefs 和 scales 包含了连续小波系数和相关的尺度。其中，coefs 为一个 64×2188 的矩阵，scales 为一个 1：64 的向量。

2.2 一维连续复小波分析

通过本节介绍，将掌握利用小波工具箱的命令行和图形工具两种方式来进行一维连续复小波分析，重点放在理解一维连续复小波分析和一维连续实小波分析的区别上。

2.2.1 一维连续复小波分析——命令行方式

以图 2-13 所示尖峰信号为例，用命令行方式进行一维连续复小波分析。

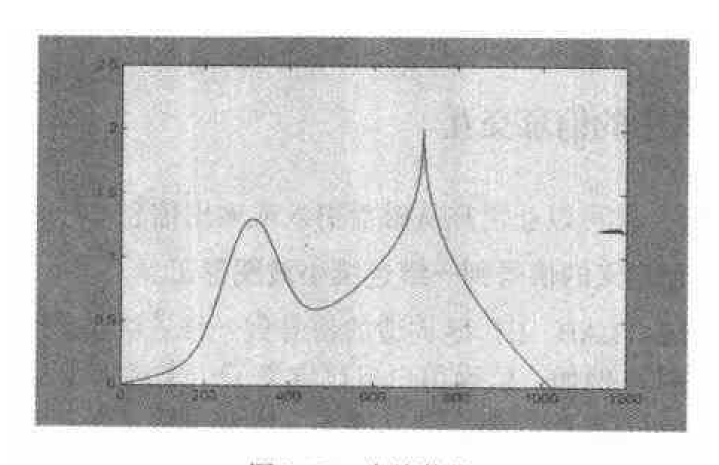


图 2-13 尖峰信号

1．装载信号

在 MATLAB 命令行窗口中输入：

```
>>load cuspamax;
>>whos
  Name          Size        Bytes       Class
  caption       1×71        142         char array
  cuspamax      1×1024      8192        double array
>>caption
  caption =
   x = linspace(0,1,1024); y = exp(-128*((x-0.3).^2))-3*(abs(x-0.7).^0.4);
```

说明：这里 caption 是包含定义信号的字符串。

2．完成连续小波变换

仍然采用 cwt 函数：

```
c = cwt(cuspamax,1:2:64,'cgau4');
```

说明：这里小波函数为 cgau4。MATLAB 6.5 中用来进行复小波分析的基本小波函数只有两个：复高斯小波 cgauwavf 和复 Morlet 小波 cmorwavf。用户可以通过命令

waveinfo('cgau');和 waveinfo('cmor')来查看其主要性质。

3．显示系数的图形表示

函数 cwt 的第四个参数可以产生复小波分析的四个变换系数：实部和虚部，模和相位，这是与第一节实小波所不同的地方。键入以下命令：

```
>>c = cwt(cuspamax,1:2:64,'cgau4','plot');
```

则相应的系数表示如图 2-14 所示，分别为实部、虚部、模和相位。

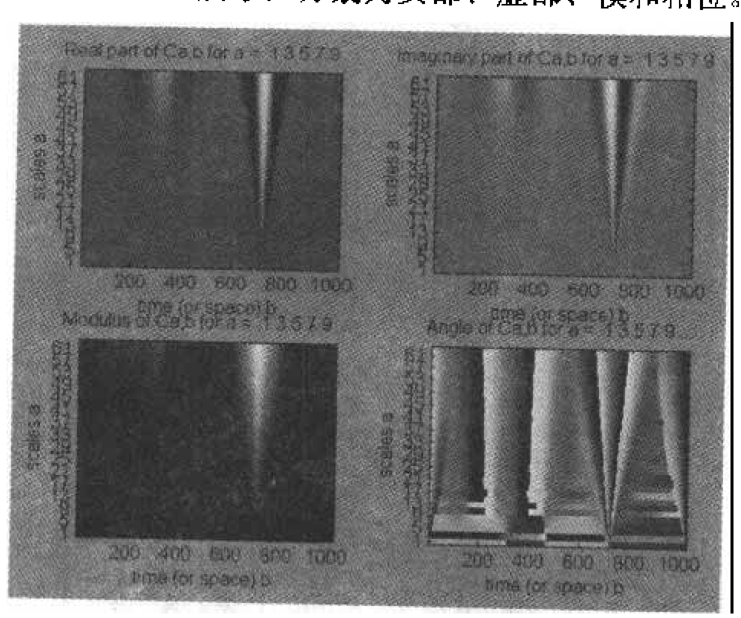


图 2-14　显示系数

2.2.2　一维连续复小波分析——图形接口方式

现在我们来采用一维连续复小波图形分析工具（Complex Continuous Wavelet 1-D）来完成上面的分析。

1．启动 Complex Continuous Wavelet 1-D 图形工具

在图 2-4 所示的小波工具箱主菜单窗口中选择 Complex Continuous Wavelet 1-D，出现如图 2-15 所示的一维连续复小波图形分析界面。

2．装载信号

选择菜单命令【File】→【Load Signal】，在出现的对话框中选取 MATLAB 6.5 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noissin.mat 文件。

3．执行一维连续复小波分析

在图 2-15 中的右上角选择基本小波为 cgaus4 和尺度为 1：2：64。单击【Analyze】按钮，出现如图 2-16 所示的分析结果。

可以看出，图 2-16 左边是系数的模，右边是系数的相位角。其余图形含义与一维连续小波图形分析工具一样，可参看上一节的说明。

用户还可以在图 2-16 右中部的 Selected 选择框中选择 Modulus，则只显示与系数的模

有关的图形，如图 2-17 所示。

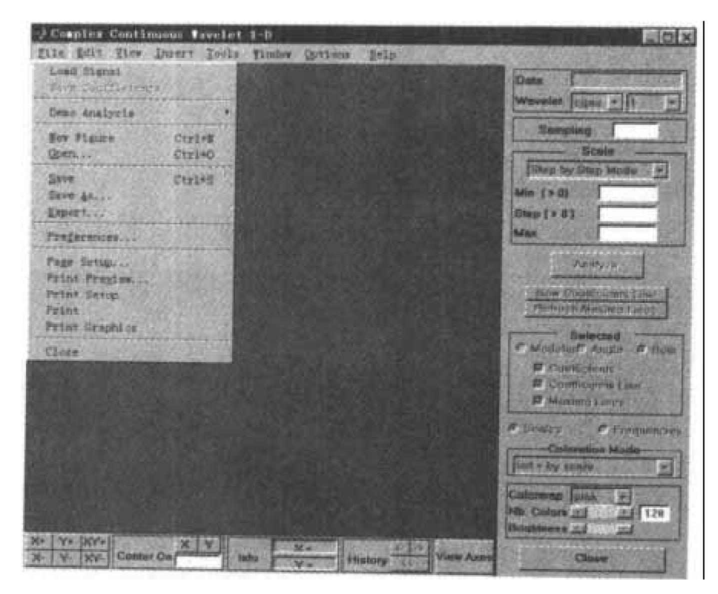


图 2-15　一维连续复小波图形分析界面

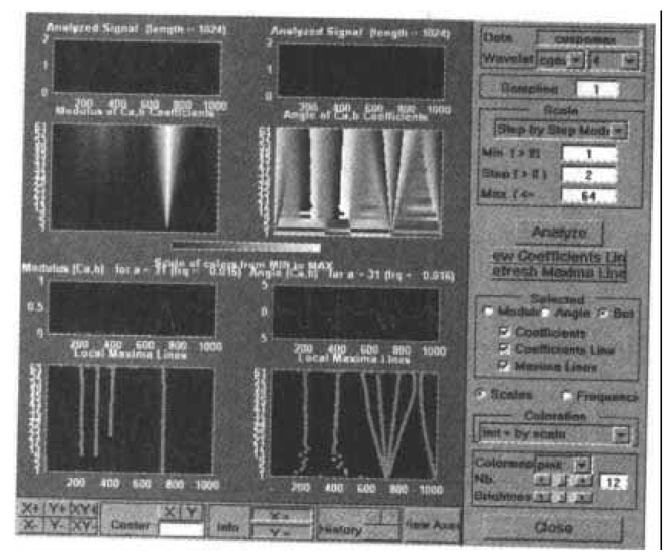


图 2-16　执行一维连续复小波分析

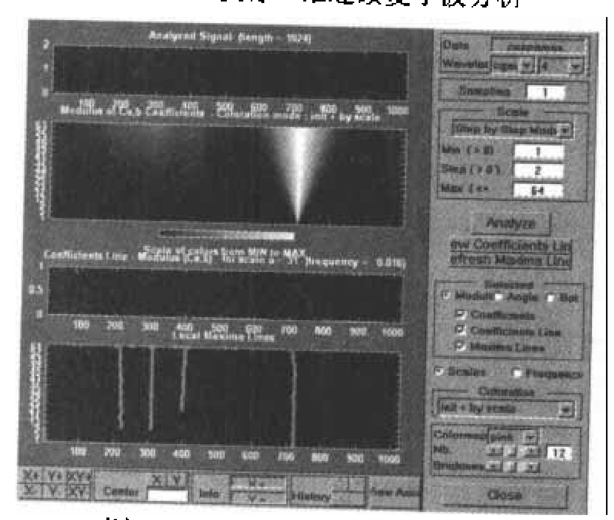


图 2-17　与系数的模有关的图形

2.2.3　图形接口方式中的信息交互

请参见 2.1.3 节的说明。惟一不同的是保存下来的变量 coefs 是一个复矩阵。

2.3　一维离散小波分析

这一节将介绍如何利用小波工具箱进行一维离散小波分析的特征。最后一部分还讨论了如何在磁盘和图形接口工具之间交换信号和系数信息。表 2-1 是这一节涉及到的小波工具箱函数，具体用法将在使用时讲解。

表 2-1　用于一维离散分析的小波工具箱函数

函数名		说明
分解函数	dwt	单尺度一维离散小波变换
	wavedec	多尺度一维小波分解（一维多分辨率分析函数）
	wmaxlev	允许的最大尺度值分解
合成重构函数	idwt	单尺度一维离散小波逆变换
	waverec	多尺度一维小波重构
	wrcoef	对一维小波系数进行单支重构
	upcoef	一维系数的直接小波重构
分解结构函数	detcoef	提取一维小波变换高频系数
	appcoef	提取一维小波变换低频系数
	upwlev	单尺度一维小波分解的重构
消噪压缩函数	ddencmp	获取消噪或压缩过程中的默认阈值
	wbmpen	以 Birge-Massart 算法设置一维或二维信号消噪的阈值
	wdcbm	以 Birge-Massart 算法设置一维信号消噪或压缩的阈值
	wdencmp	用小波进行一维或二维信号的消噪或压缩
	wden	用小波进行一维信号的自动消噪
	wthrmngr	阈值管理

2.3.1　一维离散小波分析——命令行方式

以现实生活中一个电网监测的电压信号为例，如图 2-18 所示。这个信号包含了测量时由于监测设备的故障而引入的噪声。从下面的小波分析结果可以看出能够有效地去除噪声。

1. 装载信号

在 MATLAB 命令行窗口中输入：

```
>>load leleccum;
```

设置变量：

```
>>s=leleccum(1:3920);
>>ls=length(s);
```

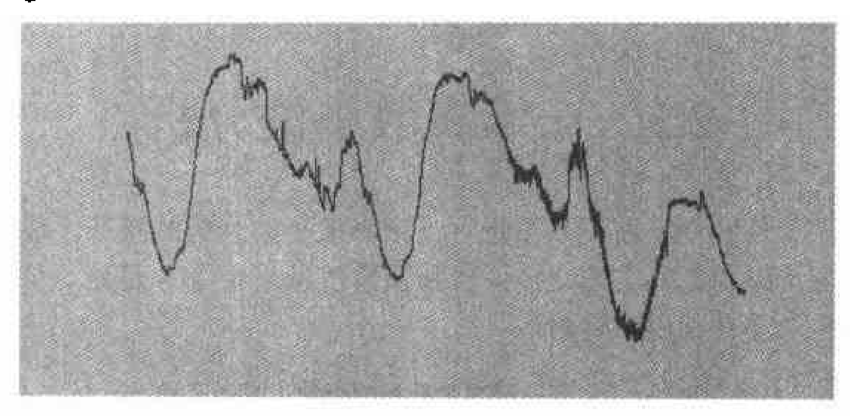


图 2-18　电网监测的电压信号

2．**完成信号的单尺度一维离散小波分解**

采用 db1 基本小波来分解信号：

```
[cA1,cD1]=dwt(s,'db1');
```

这就产生了低频系数 cA1 和高频系数 cD1。

这里对单尺度一维离散小波变换函数 dwt 做一介绍。其格式为：

① [cA1,cD1]=dwt(X,'wname')

② [cA1,cD1]=dwt(X,Lo_D,Hi_D)

说明：其中 X 为被分析的离散信号，wname 为分解所用到的小波函数，Lo_D、Hi_D 为分解滤波器，cA 和 cD 分别为返回的低频系数和高频系数向量，它们长度相等为 length（X）/2（当 length（X）为偶数时）或（length（X）＋1）/2（当 length（X）为奇数时）。

3．**从系数中重构低频和高频部分**

从第三步中产生的系数 cA1 和 cD1 构造第一层的低频和高频（A1 和 D1）系数：

```
A1=upcoef('a','cA1','db1',1,ls);
D1=upcoef('d','cD1','db1',1,ls);
```

或

```
A1 = idwt(cA1,[],'db1',l_s);
D1 = idwt([],cD1,'db1',l_s);
```

这里 upcoef 是一维系数的直接小波重构函数，其格式为：

① Y=upcoef(O,X,'wname',N)

② Y=upcoef(O,X,'wname',N，L)

③ Y=upcoef(O,X,Lo_R,Hi_R,N)

④ Y=upcoef(O,X,Lo_R,Hi_R,N,L)

⑤ Y=upcoef(O,X,'wname')

⑥ Y=upcoef(O,X,Lo_R,Hi_R)

说明：该函数用于计算向量 X 向上 N 步的重构小波系数，N 是严格的正整数。如果 O＝a，则是对低频系数进行重构，如果 O＝d，则是对高频系数进行重构。对于格式②、

④的情况，则是对向量 X 中间长度为 L 的部分进行重构。在重构的过程中，格式①、②、⑤是用小波函数进行重构，格式③、④、⑥是用重构滤波器进行重构。另外，格式⑤等价于 Y=upcoef(O,X,'wname',1)；格式⑥等价于 Y=upcoef(O,X,Lo_R,Hi_R,1)。函数 idwt 用法参见下面的说明。

4. 显示低频和高频部分

为了显示第一层的分解结果，键入：

```
subplot(1,2,1);plot(A1);title('Approximation A1')
subplot(1,2,2);plot(D1);title('Detail D1')
```

结果如图 2-19 所示。

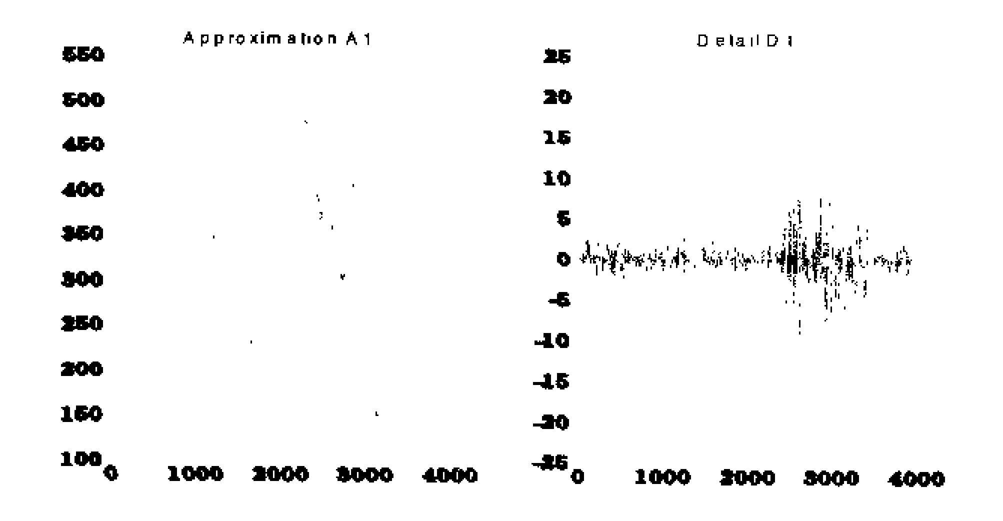


图 2-19 低频和高频部分

5. 由小波逆变换恢复信号

键入命令：

```
>>A0=idwt(cA1,cD1,'db1',ls);
```

其中 idwt 为单尺度一维离散小波逆变换函数。格式为：

```
①X=idwt(cA,cD,'wname')
②X=idwt(cA,cD,Lo_R,Hi_R)
③X=idwt(cA,cD,'wname',L)
④X=idwt(cA,cD,Lo_R,Hi_R,L)
```

说明：对格式①、③，它是用小波函数进行重构，对于格式②、④，是用重构滤波器进行重构。cA 和 cD 的长度是相等的，Lo_R 和 Hi_R 的长度是相等的。X 为重构后信号的向量。

6. 多尺度一维分解

为了完成一个三尺度的分解（仍用 db1），键入：

```
>>[C,L]=wavedec(s,3,'db1');
```

其中 wavedec 为多尺度一维小波分解函数，其格式为：

```
①[C,L]=wavedec(X,N,'wname')
②[C,L]=wavedec(X,N,Lo_D,Hi_D)
```

说明：它用小波或分解滤波器完成对信号 X 的一维多尺度分解，N 为尺度，且为严格的正整数。输出参数 C 由[cA_j，cD_j，cD_{j-1}，…，cD_1]组成，L 由[cA_j的长度，cD_j的长度，cD_{j-1}的长度，…，cD_1的长度，X 的长度]组成。例如，一个三尺度的分解结构的组织形式如图 2-20 所示。

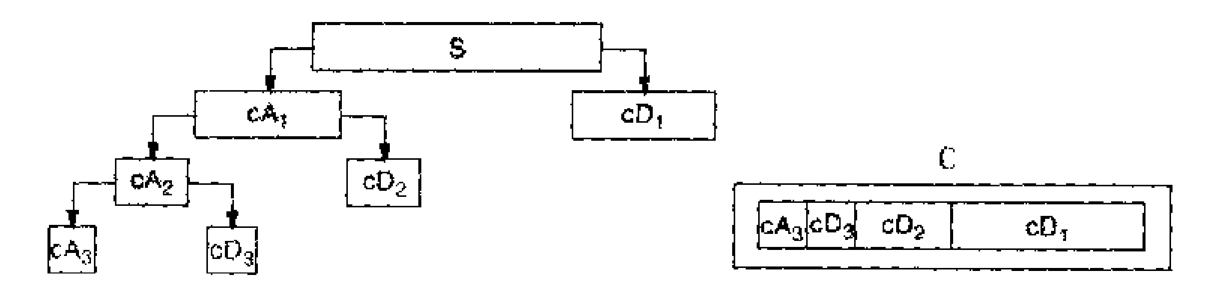


图 2-20　三尺度分解结构的组织形式

7．提取系数的低频和高频部分

为了从上面的 C 中提取第三层的低频系数，键入：

```
cA3=appcoef(C,L,'db1',3);
```

这里 appcoef 用于从小波分解结构[C,L]中提取一维信号的低频系数，格式为：

```
①A=appcoef（C,L,'wname',N）
②A=appcoef（C,L,'wname'）
③A=appcoef（C,L,Lo_R,Hi_R）
④A=appcoef（C,L,Lo_R,Hi_R,N）
```

说明：其中[C,L]为小波分解结构，wname 为小波函数，N 为尺度。格式①计算尺度 N（N 必须为一个正整数，且 0≤N≤length(L)－2）时的一维分解低频系数；格式②用于提取最后一尺度（尺度 *N*=length(*L*)－2）的小波变换低频系数；格式③、④是用滤波器 Lo_R 和 Hi_R 进行信号低频系数的提取。该滤波器一般与某一小波函数相关，通常可由 wfilters 函数得到。返回系数 A 是一个向量，向量的元素个数为 length(L)/2^N。

为了从上面的 C 中提取第 1、2、3 层的高频系数，键入：

```
cD3=detcoef(C,L,3);
cD2=detcoef(C,L,2);
cD1=detcoef(C,L,1);
或 [cD1,cD2,cD3] = detcoef(C,L,[1,2,3]);
```

这里 detcoef 用于从小波分解结构[C,L]中提取一维信号的高频系数，格式为：

```
①D=detcoef(C,L,N)
②D=detcoef(C,L)
```

说明：格式①用于提取尺度为 N（N 必须为一个正整数，且 0≤N≤length(L)－2），分解结构为[C,L]的一维分解高频系数；格式②用于提取最后一尺度（尺度 N=length(L)－2）的一维分解高频系数，该函数返回一个向量，其长度为 length(L)/2^N。

8．重构第三层的低频信号

为了从上面的 C 中重构第三层的低频信号，键入：

```
>>A3=wrcoef('a',C,L,'db1',3);
```

这里 wrcoef 是对一维信号的分解结构[C,L]用指定的小波函数或重构滤波器进行重构



的函数。其格式有：

①X=wrcoef('type',C,L,'wname',N)

②X=wrcoef('type',C,L,Lo_R,Hi_R,N)

③X=wrcoef('type',C,L,'wname')

④X=wrcoef('type',C,L,Lo_R,Hi_R)

说明：当 type＝a 时，对信号的低频部分进行重构，此时 N 可以为 0；当 type＝d 时，对信号的高频部分进行重构，此时 N 为正整数。

9．重构第 1、2、3 层的高频信号

其方法是：

```
D1=wrcoef('d',C,L,'db1',1);
D2=wrcoef('d',C,L,'db1',2);
D3=wrcoef('d',C,L,'db1',3);
```

10．显示多尺度一维分解结果

其方法是：

```
subplot(2,2,1);plot(A3);title('Approximation A3')
subplot(2,2,2);plot(D1);title('Detail D1')
subplot(2,2,3);plot(D2);title('Detail D2')
subplot(2,2,4);plot(D3);title('Detail D3')
```

结果如图 2-21 所示。

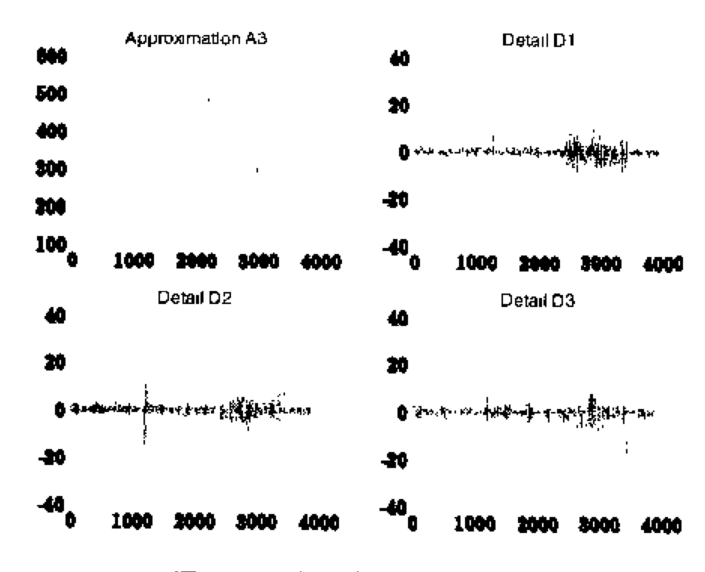


图 2-21　多尺度一维分解结果

11．重构原始信号

其方法是：

```
A0=waverec(C,L,'db1');
```

重构最大误差为：

```
err = max(abs(s-A0))
err =4.5475e-013
```

这里waverec为多尺度一维小波重构函数，用指定的小波函数或重构滤波器对小波分

解结构[C,L]进行重构，格式为：

①X=waverec(C,L,'wname')

②X=waverec(C,L,Lo_R,Hi_R)

说明：它是 wavedec 函数的逆函数，即有 X=waverec(wavedec(X,N,'wname'),'wname')。另外，X=waverec(C,L,'wname')与 X=appcoef(C,L,'wname',0)等价。格式①用小波函数进行重构，格式②用重构滤波器进行重构。

12. 信号的初步去噪

比较第三层近似信号与原始信号，如图 2-22 所示。

```
subplot(2,1,1);plot(s);title('original');axis off
subplot(2,1,2);plot(A3);title('Level 3 Approximation');axis off
```

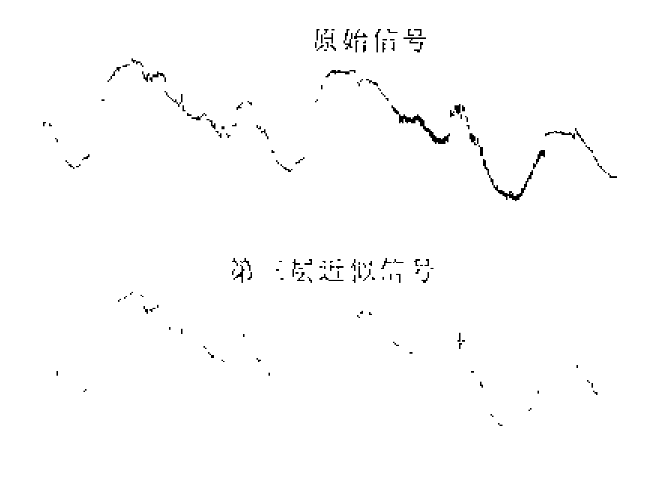


图 2-22 信号的初步去噪

利用小波对信号去噪，首先要识别出信号的哪一部分或哪些部分包含噪声，然后舍弃这些部分进行信号重构。在本例中，当越来越多的高频信息从信号中滤去时，相应的低频部分变得越来越"纯洁"，即所含的噪声越来越小。由图 2-22 可以看出，第三层的近似信号与原始信号相比噪声小了很多。当然，我们在去掉高频部分的同时也丢失了初始信号的瞬变特征。因此，更优化的去噪是阈值消噪，即只去除那些超过某一设定值的细节部分。关于信号去噪的详细讨论可参见第 3 章中的实例分析部分。

2.3.2 一维离散小波分析——图形接口方式

在这一部分，我们仍以上面电网监测的电压信号为例，但采用 MATLAB 6.5 的图形工具箱进行分析。

1. 启动一维离散小波分析图形工具

在图 2-4 小波工具箱主菜单中选择 Wavelet 1-D，出现如图 2-23 所示的一维离散小波分析图形工具。

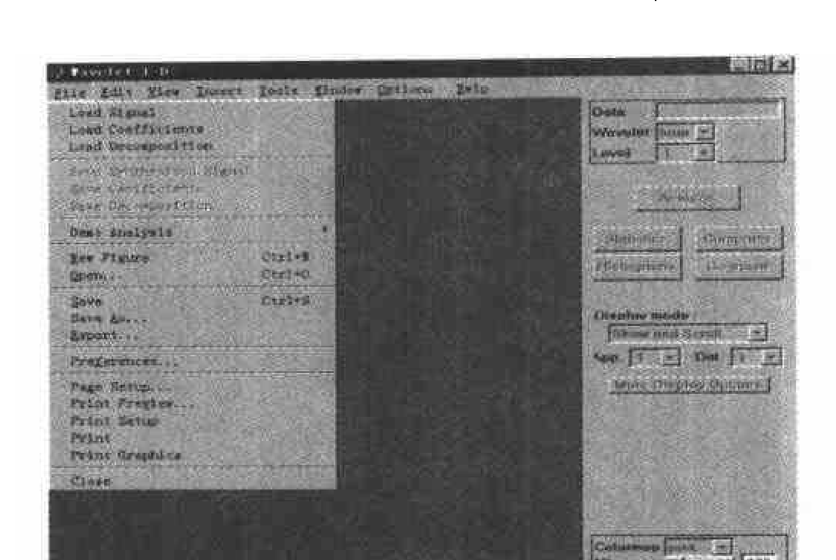


图 2-23　一维离散小波分析图形工具

2．装载信号

单击【File】→【Load Signal】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 leleccum.mat 文件。

3．完成单尺度一维离散小波变换

在图 2-23 中的右上角选择基本小波为 db1 和尺度 Level 为 1。选择好以上参数后，就可以单击【Analyze】按钮。经过短时间的计算后将出现如图 2-24 所示的分解结果。

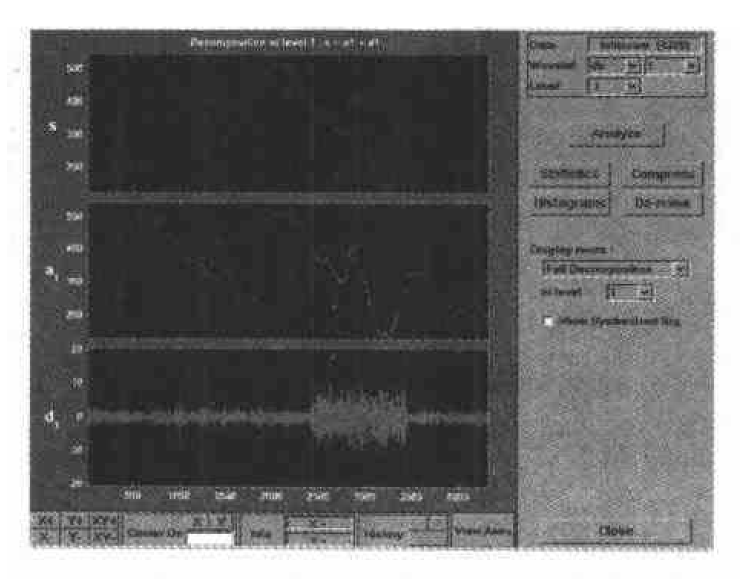


图 2-24　单尺度一维离散小波变换

4．相关细节部分放大

按住鼠标左键不放，在信号明显含有噪声的部分拖放一个矩形框。然后单击图 2-24 底部的【X+】按钮，即只在水平方向上放大。出现图 2-25 所示的放大了的信号和对应的系数图，读者可以清楚地看出噪声信号的特征。其他的放大控制按钮见图 2-25 的底部。其中可以通过 History 旁的左右箭头回到前一次或下一次的放大情形。

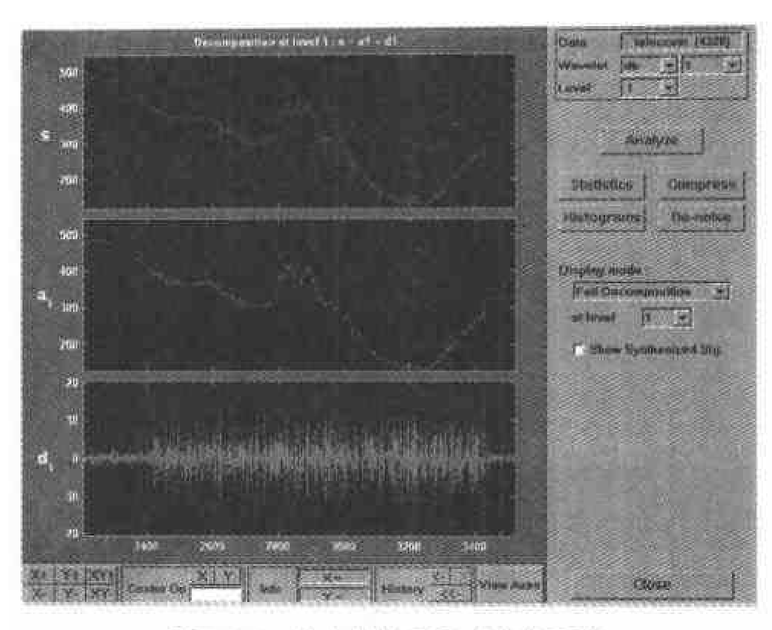


图 2-25　放大的信号和对应的系数

5．多尺度一维分解

重新选择图 2-23 中右上角的 Level 值为 3，Wavelet 仍为 db1，单击【Analyze】按钮来完成前面在命令行方式中实现的三尺度分解。结果如图 2-26 所示。

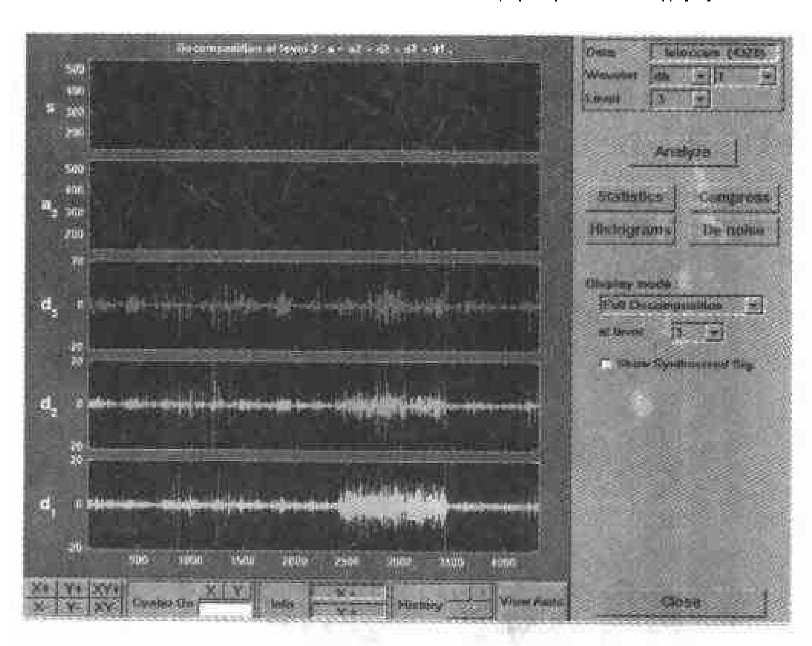


图 2-26　三尺度分解

6．选择信号分解显示模式

在默认情况下，一个信号用小波分解后，显示模式是完全分解模式（Full Decomposition）。除此之外，MATLAB 6.5 小波工具箱还提供了另外几种显示模式，用户可以在如图 2-26 右侧所示的显示模式选择框（Display mode）中进行选择。共有以下 6 种模式：

（1）完全分解模式（Full Decomposition）

在该模式下，信号分解的高频部分和低频部分可以全部显示出来，具体显示到第几层，

由 at level 选择框决定。选中 Show Synthesized Sig 标记框，可与原始信号在同一个坐标系中显示合成后的信号。

（2）显示滚动模式一（Show and Scroll）

在该模式下，显示三个窗口。无论将信号分解到第几层，每次只分别显示由 App.和 Det.选择框决定的一个低频系数和一个高频系数。系数的显示设置可通过单击【More Display Options】按钮，在弹出的窗口进行。

（3）显示滚动模式二（Stem Cfs）

该模式与以上的显示滚动模式非常相似，惟一不同之处是在第三个窗口中以杆状图而不是以染色模块来显示小波分解系数。

（4）分离模式（Separate Mode）

和完全分解模式的显示类似，但将高频部分和低频部分的系数分成两列来显示。具体显示第几层的系数，由单击【More Display Options】按钮后弹出的 More Detail Option 窗口来设置。在该窗口中，选择要显示的层数后单击【Apply】按钮，就可以看到系数已按相应的风格显示出来了。

（5）叠加模式（Superimpose Mode）

和显示滚动模式下的显示结果基本相同，不同之处在于叠加模式是将低频的各层系数都叠加在一起在一个方框中显示，将高频的各层系数也都叠加在一起在一个方框中显示，各层系数用不同颜色的曲线表示。可以通过单击【More Display Options】按钮后设置各种参数来决定显示的具体内容。

（6）树模式（Tree Mode）

在该模式下，显示三个方框，最左边的用于显示小波分解的树结构；右上角的方框用于显示原始信号；右下角的方框显示内容由用户自己选择。用户用鼠标单击小波分解树中的任一个节点（s 或 a_i 或 d_i），则在右下角的方框中显示该节点系数的重构信号。

图 2-27 是各种显示模式的对比。用户还可以通过选择【Options】→【Default Display Mode】菜单命令来改变默认显示模式。

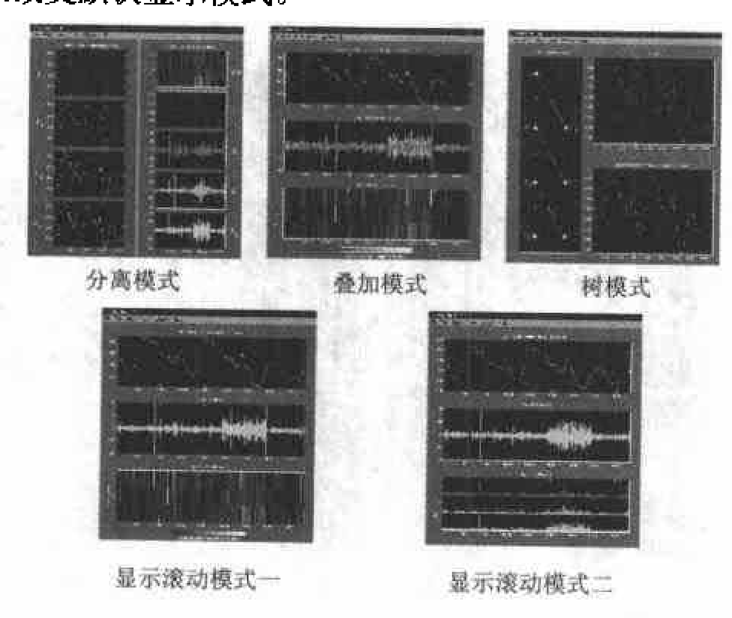


图 2-27　显示模式



从 Wavelet 1-D 工具界面中打开的压缩和消噪窗口将继承 Wavelet 1-D 中设置的系数显示属性。

7．信号去噪

图形工具的预定义阈值去噪方式可以很容易地将噪声从信号中去除。单击信号小波分解后窗口右边出现的【De-noise】按钮，出现相应的 Wavelet 1-D De-noising 窗口，如图 2-28 所示。

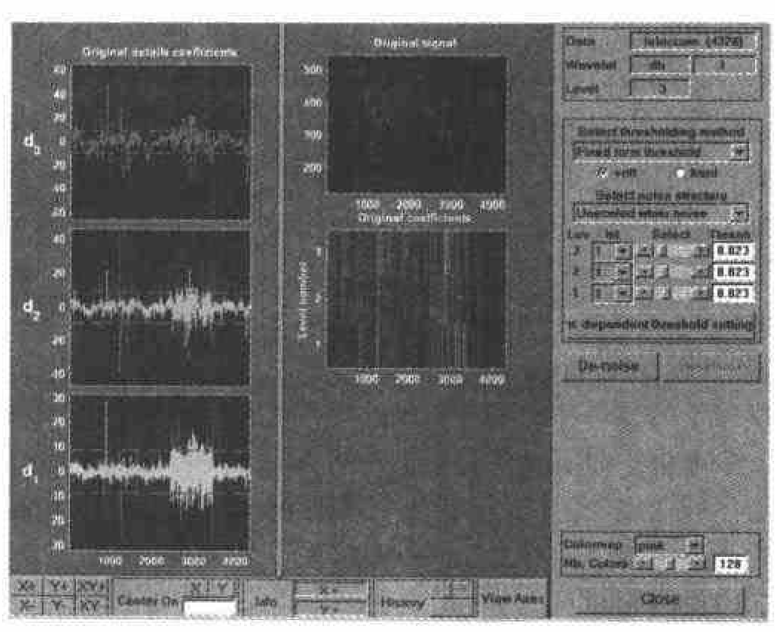


图 2-28　信号去噪

我们选择软阈值（soft thresholding）和尺度未知的白噪声（unscaled white noise），然后单击【De-noise】按钮。消噪效果如图 2-29 所示，大家可以选择自己感兴趣的部分放大细看。

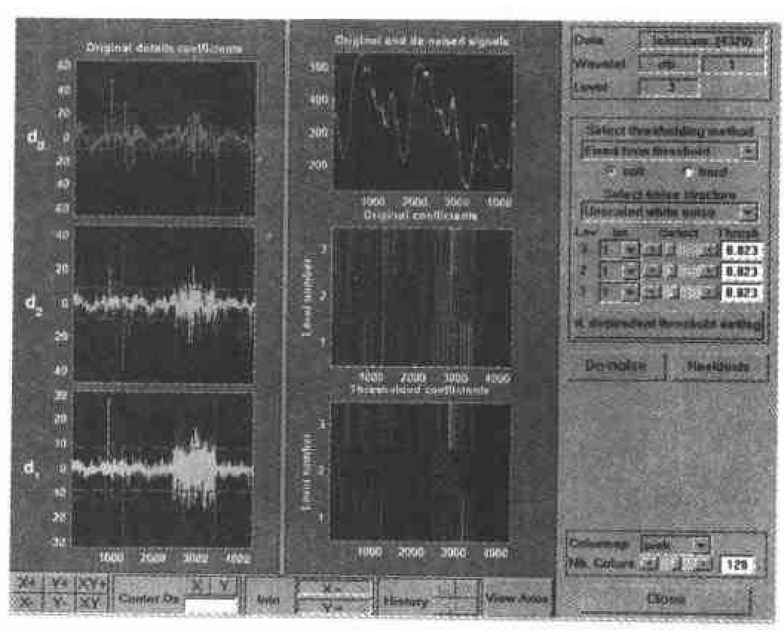


图 2-29　消噪效果

由于消噪（De-noise）和压缩（Compress）窗口不能同时打开，所以先关闭消噪窗口以继续下面的分析。此时出现 Update Synthesized Signal 对话框时，选择【No】按钮。

8．信号压缩

小波图形工具箱提供了自动阈值（Automatic thresholding）和手动阈值（Mannual thresholding）两种方式进行数据压缩。单击【Compress】按钮，出现图 2-30 所示数据压缩窗口。

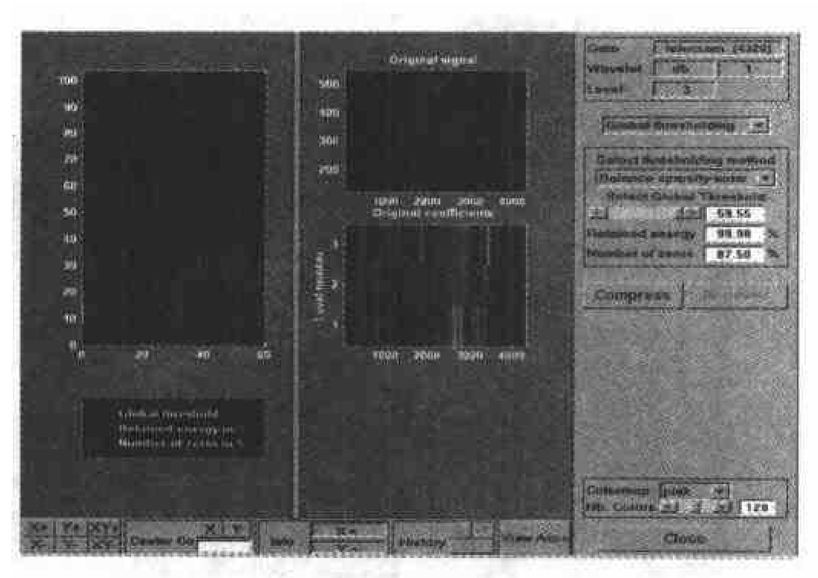


图 2-30　数据压缩窗口

当选择自动阈值时，阈值是一个全局阈值，可在 Select Global Threshold 选择框中拖动滚动条或在文本框中输入数值来选择该阈值，也可在左边的 Automatic thresholding 显示方框中用鼠标拖动绿色的阈值虚线来选择。当进行手动系数阈值量化时，用户可以对高频系数的每一层都选择不同的阈值，每层阈值的选择方法和上述类似。选择好阈值后，单击【Compress】按钮，压缩信号（黄色）和原始信号（红色）叠加显示在图 2-31 中（彩色效果图见彩插 2）。

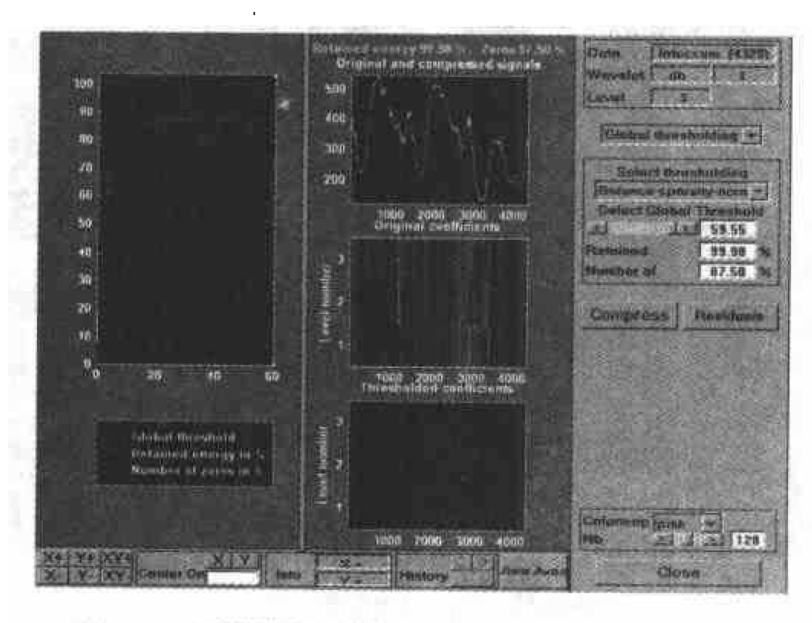


图 2-31　压缩信号（黄色）和原始信号（红色）叠加

可以看出，压缩后大部分噪声被去除了，但保留了原信号的 99.98%的能量，置 0 系数的百分比为 87.50%。

9．显示其他信息

在 Wavelet 1-D Compress 工具中单击【Residuals】按钮，出现 More on Residuals for Wavelet 1-D Copress 窗口，如图 2-32 所示。可以看出，除均值、模式、中值、误差、标准差等统计量外，时间序列图（自相关和自谱）及频域分布图也被显示。

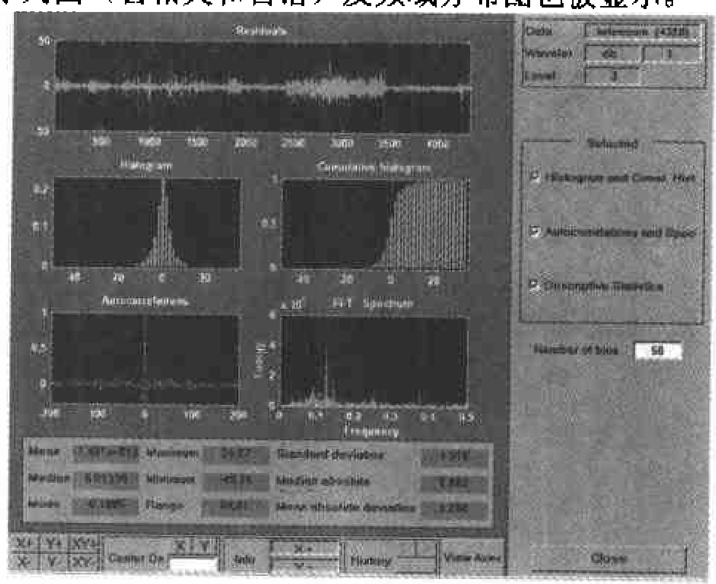


图 2-32　其他信息

10．信号统计

从 Wavelet 1-D 工具中单击【Statistics】按钮，出现如图 2-33 所示的一维离散小波统计窗口。

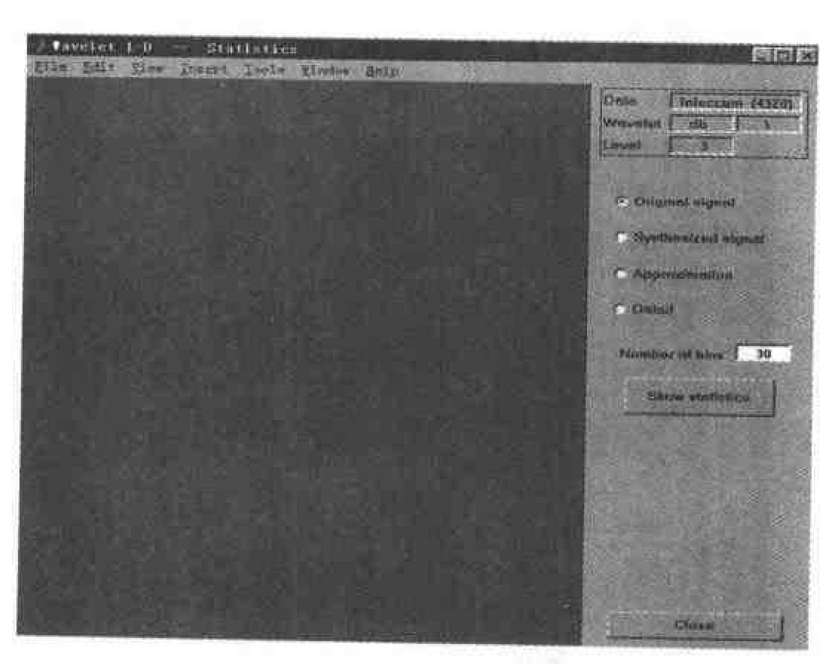


图 2-33　一维离散小波统计窗口

窗口右边有四个单选按钮，分别是 Orignal signal、Synthesized signal、Approximation 和 Detail，选择其中某个按钮，则在左边的显示区域中显示相应信号的统计结果，包括文本方式显示的计算值、以图形方式显示的直方图和以求和形式显示的直方图。当选中 Approximation 或 Detail 后，在窗口中部会出现 Approximation at 或 Detail at 的选择框，让用户具体选择要显示的低频系数或高频系数的层次。柱条数（Number of bins）文本框可以让用户输入两种直方图中显示的柱条数量。这里我们选择 Approximation 单选按钮和 Level 1，然后单击【Show statistics】按钮，统计结果如图 2-34 所示。

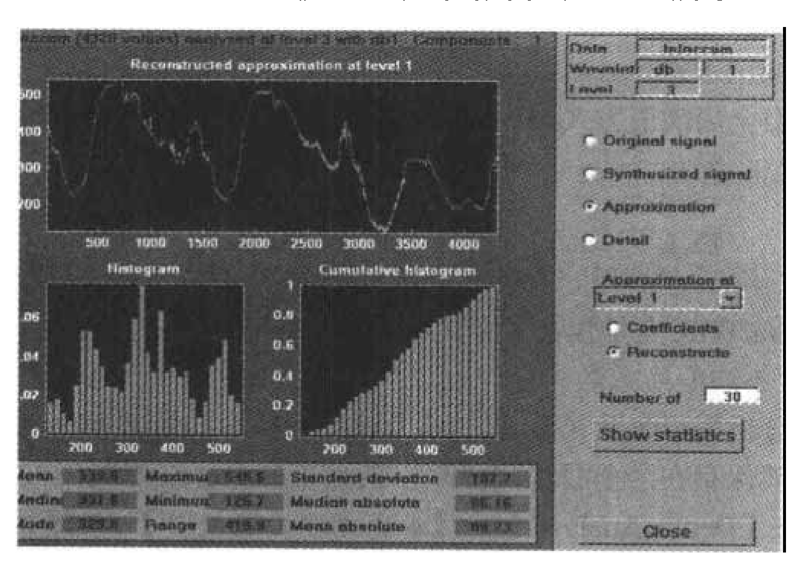


图 2-34　统计结果

2.3.3　图形接口方式中的信息交互

一维离散小波图形工具可以让用户从磁盘引入或输出信息，如图 2-23 所示的【File】下拉菜单。

1．从一维离散小波图形工具保存信息到磁盘

用户可以将一维离散小波图形工具产生的合成信号、系数和分解后的各部分保存到磁盘，这些信息以后还可以被一维离散小波图形工具重新利用。

（1）保存合成信号

用户可以将一维离散小波图形工具处理后的信号以 MAT 格式的文件保存起来。例如，首先调入一个图形工具自带的演示分析信号，选择菜单命令【File】→【Demo Analysis】→【with db3 at level 5】→【Sum of sines】，然后执行对其执行压缩或去噪。当关闭压缩或去噪窗口时，在出现的 update the synthesized signal 对话框中单击【Yes】按钮，再进入图形工具主菜单选择【File】→【Save Synthesized Signal】命令。在出现的对话框中选择保存路径和文件名，这里我们取名为 synthsig.mat。为了将该信号装载进用户工作区，只需要在命令提示符后键入 load synthsig。

当合成信号是通过非全局阈值方法获得时，其保存的结构如下：

```
>>whos
>>Name        Size        Bytes        Class
```

synthsig	1×1000	8000	double array
thrParams	1×5	580	cell array
wname	1×3	6	char array

可以注意到，压缩或消噪的参数由小波基函数（wname）给出，而与层次相关的阈值信息保存在长度为 5 的 thrParams 变量里。

i 从 1 到 5，thrParams{i}包含了阈值间隔的上下限及阈值，例如对 level 1：

```
>>thrParams{1}
ans =
1.0e+03 *
0.0010 1.0000 0.0014
```

当合成信号是由全局阈值方式得到时，其保存结构为：

Name	Size	Bytes	Class
synthsig	1×1000	8000	double array
valTHR	1×1	8	double array
wname	1×3	6	char array

说明：这里的 valTHR 变量包含了全局阈值，valTHR＝1.2922。

（2）保存离散小波变换系数

一维离散小波分析图形工具可让用户将 DWT 变换后的系数保存到磁盘。工具箱以用户选择的文件名将其保存在当前目录。还是以其自带的 Demo 为例：

选择【File】→【Demo Analysis】→【with db1 at level 5】→【Cantor curve】菜单命令。

当以 cantor.mat 保存小波变换系数后，再将其装载进工作区：

```
>>load cantor
>>whos
```

Name	Size	Bytes	Class
coefs	1×2190	17520	double array
longs	1×7	56	double array
thrParams	0×0	0	double array
wname	1×3	6	char array

说明：这里变量 coefs 包含了离散小波变换系数。更精确地讲，在上面的例子里 coefs 是一个 1×2190 的向量，longs 包含了系数的每一部分的尺度。

变量 wname 是基小波名，thrParams 是一个空向量（因为这里合成信号不存在）。

（3）保存分解后的信号

一维离散小波分析可以在当前的目录下以 wa1 扩展名来保存分解后的信号。还是以其自带的 Demo 为例：

选择菜单命令【File】→【Demo Analysis】→【with db3 at level 5】→【Sum of sines】。

为了保存这次的分析结果，在主菜单中选择【File】→【Save Decomposition】。

然后在出现的对话框中键入文件名 wdecexd1d。然后将其装载进工作区：

```
>>load wdecex1d.wa1 -mat
>>whos
```

Name	Size	Bytes	Class
coefs	1×1023	8184	double array

data_name	1×6	12	char array
longs	1×7	56	double array
thrParams	0×0	0	double array
wave_name	1×3	6	char array



保存选项在一维离散小波分析图形工具中的消噪和压缩窗口依然有效，这样用户就不必回到一维离散小波分析图形工具的主窗口就可以直接在消噪或压缩窗口保存相应的信息。

2. **装载信息到一维离散小波分析图形工具**

用户可以装入信号数据或小波分解系数数据或小波分解后的信号数据到图形工具中进行分析。这些信息可能是先前从图形工具中保存到磁盘，然后又在工作区进行过处理或者是用户通过命令行方式产生的。不管是何种情形，用户必须严格遵守一维小波工具所要求的文件格式和数据结构，否则就会产生出错信息。

（1）装载信号

例如，假定用户定义了一个名为 warma 的信号以利用 Wavelet 1-D 工具进行分析，首先执行以下命令：

```
>>save warma
>>sizewarma=size(warma)
>>sizewarma=
1000
```

然后就可以通过菜单命令【File】→【Load Signal】装入该信号了。

（2）装载一维离散小波变换系数

在装载之前，首先要有包含至少 coefs 和 longs 两个变量的 MAT 格式文件。其格式和含义请参见前面的说明。图 2-35 是某个具体信号的示意图。

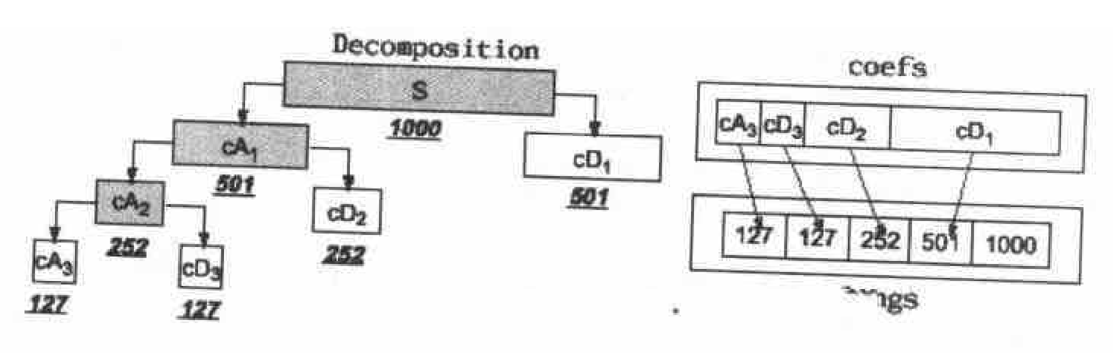


图 2-35　某具体信号示意图

当在工作区编辑或构造好合适的信号后，键入：

```
>>save myfile coefs longs
```

然后在菜单中选择【File】→【Load Coefficients】命令，选择 myfile 即可。

（3）装载一维离散小波分解后的信号

在装载之前，要以.wa1 为扩展名将有关数据保存为 MAT 格式的文件。该数据文件必须包含 coefs、wave_name 和 longs 三个变量，变量 data_name 是可选的。

当在工作区编辑或构造好合适的信号后，键入：

```
>>save myfile coefs longs wave_name
```

然后再选择菜单命令【File】→【Load Decomposition】将 myfile 装入。

2.4 二维离散小波分析

本节将介绍如何利用小波工具箱进行二维离散小波分析，涉及到的用于二维图像分析的小波工具箱函数如表 2-2 所示。

表 2-2 用于二维图像分析的小波工具箱函数

函数名		说明
分解函数	dwt2	单尺度二维离散小波变换
	wavedec2	多尺度二维小波分解（二维多分辨率分析函数）
	wmaxlev	允许的最大尺度分解
合成重构函数	idwt2	单尺度逆二维离散小波变换
	waverec2	多尺度二维小波重构
	wrcoef2	对二维小波系数进行单支重构
	upcoef2	对二维小波分解的直接重构
分解结构函数	detcoef2	提取二维小波分解高频系数
	appcoef2	提取二维小波分解低频系数
	upwlev2	二维小波分解的单尺度重构
消噪压缩函数	ddencmp	获取消噪或压缩过程中的默认阈值
	wbmpen	以 Birge-Massart 算法设置一维或二维信号消噪的阈值
	wdcbm2	以 Birge-Massart 算法设置二维信号消噪或压缩的阈值
	wdencmp	用小波进行一维或二维信号的消噪或压缩
	wthrmngr	阈值管理

2.4.1 二维离散小波分析——命令行方式

本节中我们将应用二维小波分析方法来有效地压缩一个图像而不失其清晰度。

1. 装载图像信号

在 MATLAB 命令行窗口中输入：

```
>>load wbarb;
>>whos
  Name      Size         Bytes     Class
  X         256×256      524288    double array
  map       192×3        4608      double array
```

2. 显示图像

在 MATLAB 命令行窗口中输入：

```
>>image(X); colormap(map); colorbar;
```

相应图像显示如图 2-36 所示。

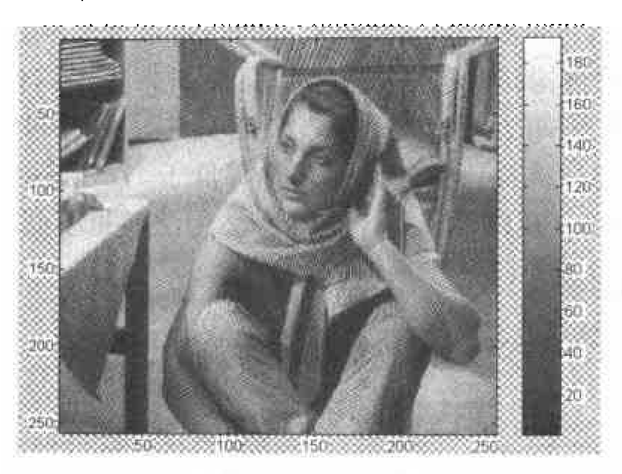


图 2-36 显示图像

3．完成图像的单尺度小波分解

这里采用 bior3.7 基小波：

```
[cA1,cH1,cV1,cD1] = dwt2(X,'bior3.7');
```

这里 dwt2 是单尺度二维离散小波变换函数，其格式为：

① [cA,cH,cV,cD]=dwt2(X, 'wname')

② [cA,cH,cV,cD]=dwt2(X,Lo_D,Hi_D)

说明：X 是待分析的离散信号，wnane 为分解所用到的小波函数，Lo_D、Hi_D 为分解滤波器，cA、cH（水平）、cV（垂直）和 cD（对角线）分别为返回的低频系数和高频系数向量。

4．从系数中重构低频和高频部分

从第三步中产生的系数 cA1、cH1、cV1 和 cD1 构造第一层的低频和高频部分（A1、H1、V1 和 D1）：

```
A1 = upcoef2('a',cA1,'bior3.7',1);
H1 = upcoef2('h',cH1,'bior3.7',1);
V1 = upcoef2('v',cV1,'bior3.7',1);
D1 = upcoef2('d',cD1,'bior3.7',1);
```

或者

```
sx = size(X);
A1 = idwt2(cA1,[],[],[],'bior3.7',sx);
H1 = idwt2([],cH1,[],[],'bior3.7',sx);
V1 = idwt2([],[],cV1,[],'bior3.7',sx);
D1 = idwt2([],[],[],cD1,'bior3.7',sx);
```

这里 upcoef2 是二维小波分解的直接重构函数，其格式为：

① Y=upcoef2(O,X,'wname',N,S)

② Y=upcoef2(O,X,Lo_R,Hi_R,N,S)

③ Y=upcoef2(O,X,'wname',N)

④ Y=upcoef2(O,X,Lo_R,Hi_R,N)

⑤ Y=upcoef2(O,X,'wname')

⑥ Y=upcoef2(O,X,Lo_R,Hi_R)

说明：该函数对第 N 层的小波分解系数进行重构，N 是严格的正整数。如果 O='a'，则是对低频系数进行重构，如果 O='h'（或'v'，'d'），则是对水平方向（或垂直方向或对角线方向）的高频系数进行重构。对于格式①、②的情况，则是对向量 X 进行重构并返回中间长度为 *S* 的部分。在重构的过程中，格式①、③、⑤是用小波函数进行重构，格式②、④、⑥是用重构滤波器进行重构。另外，格式⑤等价于 Y=upcoef2(O,X,'wname',1)，格式⑥等价于 Y=upcoef2(O,X,Lo_R,Hi_R,1)。函数 idwt2 用法参见下面的说明。

5. 显示低频和高频部分

例程 2-1 是为了显示第一层的分解结果。

例程 2-1

```
colormap(map);
subplot(2,2,1); image(wcodemat(A1,192));
title('Approximation A1')
subplot(2,2,2); image(wcodemat(H1,192));
title('Horizontal Detail H1')
subplot(2,2,3); image(wcodemat(V1,192));
title('Vertical Detail V1')
subplot(2,2,4); image(wcodemat(D1,192));
title('Diagonal Detail D1')
```

结果如图 2-37 所示。

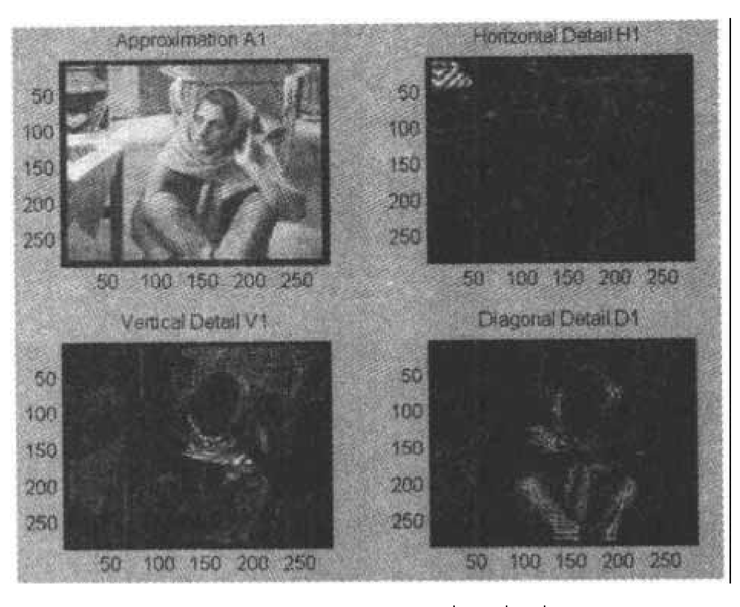


图 2-37　显示低频和高频部分

6. 由小波逆变换恢复原图像信号

键入命令：

```
Xsyn = idwt2(cA1,cH1,cV1,cD1,'bior3.7');
```

其中 idwt2 为单尺度二维离散小波逆变换函数。格式为

①X=idwt2(cA,cH,cV,cD,'wname')

②X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R)

③X=idwt2(cA, cH,cV,cD,'wname',S)

④X=idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)

说明：对格式①、③，它是用小波函数进行重构，对于格式②、④，是用重构滤波器进行重构。cA 和 cD 的长度是相等的，Lo_R 和 Hi_R 的长度是相等的。X 为重构后信号的向量。对于格式①、③，它是用小波函数进行重构，对于格式②、④，它是用重构滤波器进行重构。Lo_R 和 Hi_R 的长度是相等的。返回向量 X 为单尺度重构后的信号的低频系数。

7. 图像的多尺度二维小波分解

仍然采用 bior3.7 来完成多尺度二维小波分解：

```
[C,S] = wavedec2(X,2,'bior3.7');
```

其中 wavedec2 为多尺度二维小波分解函数，其格式为：

① [C,S]=wavedec2(X,N,'wname')

② [C,S]=wavedec2(X,N,Lo_D,Hi_D)

说明：格式①用小波进行二维分解，格式②用低通分解滤波器（Lo_D）和高通分解滤波器（Hi_D）进行二维分解。返回参数为分解结构[C,S]，其中：

C=[A(N) | H(N) | V(N) | D(N) | … H(N-1) | V(N-1) | D(N-1) | … | H(1) | V(1) | D(1)]

说明：向量 A 为低频系数，向量 H 为水平高频系数，向量 V 为垂直高频系数，向量 D 为对角线高频系数。每个向量以列的方向存储在矩阵 C 中。

矩阵 S 为：

S(1,:) =尺度 N 的低频系数长度

S(i,:) =尺度为 N-i+2 的高频系数的尺度，i = 2, …,N+1

S(N+2,:) = size(X)

8. 提取系数的低频和高频部分

为了从上面的 C 中提取第二层的低频系数，键入：

```
cA2 = appcoef2(C,S,'bior3.7',2);
```

这里 appcoef2 用于从多尺度二维小波分解结构[C,S]中提取二维信号的低频系数，格式为：

① A=appcoef2(C,S,'wname',N)

② A=appcoef2(C,S,'wname')

③ A=appcoef2(C,S,Lo_R,Hi_R)

④ A=appcoef2(C,S,Lo_R,Hi_R,N)

说明：其中[C,S]为小波分解结构，wname 为小波函数，N 为尺度。格式①计算尺度 N（N 必须为一个正整数，且 0≤N≤length(S)－2）时的二维分解低频系数；格式②用于提取最后一尺度（尺度 N＝length(S)－2）的小波变换低频系数；格式③、④是用滤波器 Lo_R 和 Hi_R 进行信号低频系数的提取。

为了从上面的 C 中提取第 1、2 层的高频系数，键入：

```
cH2 = detcoef2('h',C,S,2);
cV2 = detcoef2('v',C,S,2);
cD2 = detcoef2('d',C,S,2);
```

```
cH1 = detcoef2('h',C,S,1);
cV1 = detcoef2('v',C,S,1);
cD1 = detcoef2('d',C,S,1);
或  [cH2,cV2,cD2] = detcoef2('all',C,S,2);
[cH1,cV1,cD1] = detcoef2('all',C,S,1);
```

说明：这里 detcoef2 用于从分解机构[C,S]中提取二维小波变换的高频系数。该函数返回一个向量，其长度为 length(S)/2^N。

9. 重构第二层的低频信号

为了从上面的 C 中重构第二层的低频信号，键入：

```
A2 = wrcoef2('a',C,S,'bior3.7',2);
```

这里 wrcoef2 是对二维信号的分解结构[C,S]用指定的小波函数或重构滤波器进行单支重构的函数。其格式有：

```
①X=wrcoef2('type',C,S,'wname',N)
②X=wrcoef2('type',C,S,Lo_R,Hi_R,N)
③X=wrcoef2('type',C,S,'wname')
④X=wrcoef2('type',C,S,Lo_R,Hi_R)
```

说明：当 type=a 时，对信号的低频部分进行重构，此时 N 可以为 0；当 type=h（或 v、d）时，对信号的水平（或垂直、对角线）的高频部分进行重构，此时 N 为正整数。

10. 重构第 1、2 层的高频信号

键入：

```
H1 = wrcoef2('h',C,S,'bior3.7',1);
V1 = wrcoef2('v',C,S,'bior3.7',1);
D1 = wrcoef2('d',C,S,'bior3.7',1);
H2 = wrcoef2('h',C,S,'bior3.7',2);
V2 = wrcoef2('v',C,S,'bior3.7',2);
D2 = wrcoef2('d',C,S,'bior3.7',2);
```

11. 显示多尺度二维分解结果

请参考例程 2-2。

例程 2-2

```
colormap(map);
subplot(2,4,1);image(wcodemat(A1,192)),
title('Approximation A1')
subplot(2,4,2);image(wcodemat(H1,192));
title('Horizontal Detail H1')
subplot(2,4,3);image(wcodemat(V1,192));
title('Vertical Detail V1')
subplot(2,4,4);image(wcodemat(D1,192));
title('Diagonal Detail D1')
subplot(2,4,5);image(wcodemat(A2,192));
title('Approximation A2')
subplot(2,4,6);image(wcodemat(H2,192));
```

```
title('Horizontal Detail H2')
subplot(2,4,7);image(wcodemat(V2,192));
title('Vertical Detail V2')
subplot(2,4,8);image(wcodemat(D2,192));
title('Diagonal Detail D2')
```

结果如图 2-38 所示。

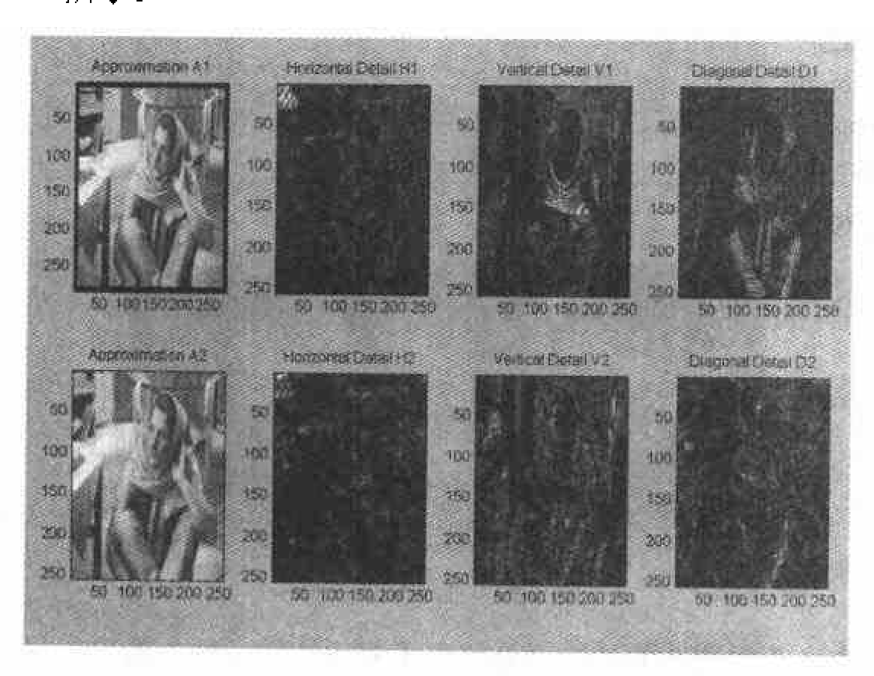


图 2-38　多尺度二维分解结果

12．重构原始图像信号

键入：

```
>>X0 = waverec2(C,S,'bior3.7');
```

这里 waverec2 是用指定的小波函数或重构滤波器在小波分解结构[C,S]上进行多尺度二维小波重构，其格式为：

①X=waverec2(C,S,'wname')

②X=waverec2(C,S,Lo_R,Hi_R)

说明：它是 wavedec2 函数的逆函数，即有 X=waverec2(wavedec2(X,N,'wname'),'wname')。另外，X=waverec2(C,S,'wname')与 X=appcoef2(C,S,'wname',0)等价。格式①用小波函数进行重构，格式②用重构滤波器进行重构。

13．压缩图像

为了压缩图像 X，我们用 ddencmp 来计算默认参数，用 wdencmp 来完成实际的压缩：

```
[thr,sorh,keepapp] = ddencmp('cmp','wv',X);
[Xcomp,CXC,LXC,PERF0,PERFL2] =wdencmp('gbl',C,S,'bior3.7',2,thr,sorh,keepapp);
```

14．显示压缩后的图像

将压缩后的图像与原始图像同时显示，如例程 2-3 所示。

例程 2-3

```
colormap(map);
subplot(121);
image(X);
title('Original Image');
axis square
subplot(122);
image(Xcomp);
title('Compressed Image');
axis square
```

结果如图 2-39 所示。

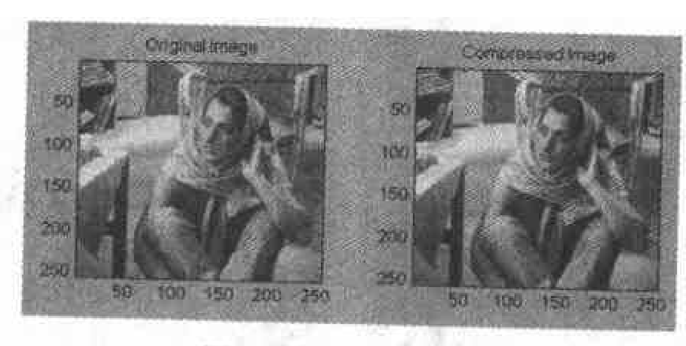


图 2-39 压缩后的图像

可通过以下命令查看置 0 的系数百分比和压缩后的能量损失情况：

```
>>PERF0
PERF0 =   49.8076      %置 0 系数百分比
>>PERFL2
PERFL2 =   99.9817     %压缩后保存了 99.9817%的能量
```

可以看出，尽管压缩后的图像只由将近一半的非 0 小波系数重构而成，但在图像的清晰度方面看不出明显的区别。

这里只是给读者一个感性的认识，关于图像压缩的详细介绍可参见后面的相关部分。

2.4.2 二维离散小波分析——图形接口方式

下面我们采用 MATLAB 6.5 的小波分析图形工具箱来完成上面的分析。

1．启动二维离散小波分析图形工具

在图 2-4 小波工具箱主菜单中选择 Wavelet 2-D，出现图 2-40 所示的二维离散小波分析图形工具。

2．装载信号

在图 2-40 中单击【File】→【Load Signal】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 wbarb.mat 文件。

3．分析图像

在图 2-40 中的右上角选择基本小波为 bior3.7 和尺度数 Level 为 2。选择好以上参数

后，就可以单击【Analyze】按钮。经过短时间的计算后将出现如图 2-41 所示的分解结果。

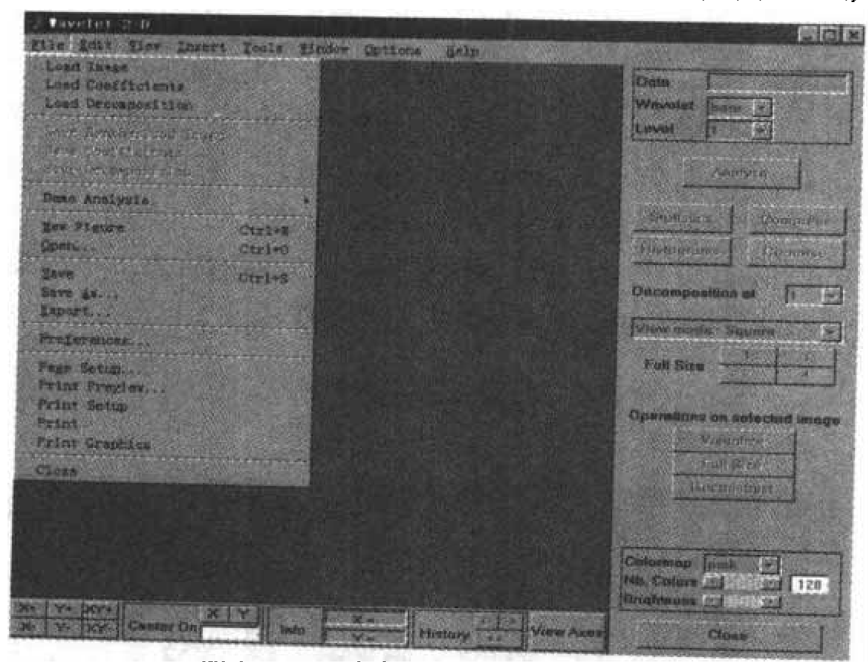


图 2-40 一维离散小波分析图形工具

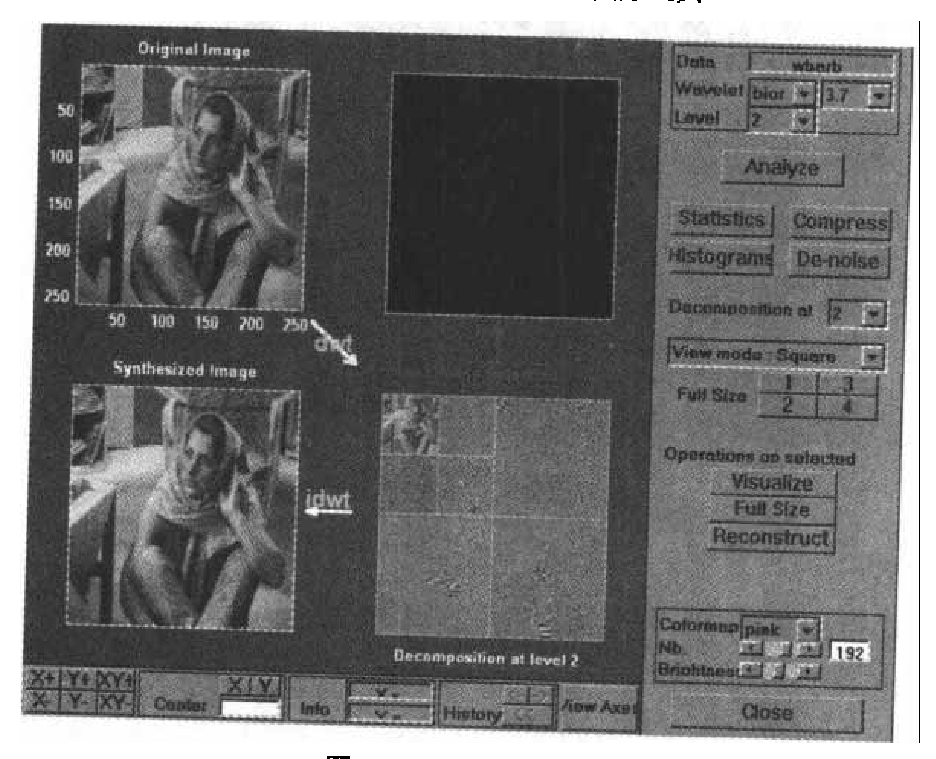


图 2-41 图像的分解

4. 方块图方式

在图 2-41 的显示模式（View mode）中可以选择小波分解后非图像显示方式，一种是方块图（Square）方式，它将分解后的图像以方块叠加的形式显示，另一种是树（Tree）模式，它将分解后的图像以树的层次形式显示。这两种方式只是形式的不同，它们都能完全表征图像的信息。二维离散小波图形工具默认的是方块图方式，如图 2-41 所示。左上方是原始图像，下面是分解后合成的图像，右下方显示的是分解的低频系数和所有的水平、

垂直和对角线高频系数。右上方可以显示用户想查看的分解的任一部分。

在图 2-41 的右下方的图中单击分解后的某一子图像，则该图像周围出现一个绿色的边框，表示该图像被选中。这样图 2-41 中的图像操作（Operations on selected image）选择栏将变得有效，用户可以选择某种操作。该栏共有三种操作：Visualize（显示）、Full Size（全屏）和 Reconstruct（重构）。单击【Visualize】按钮，可以在右上方的空白图像框中显示该图像；单击【Full Size】按钮，可以将该图像在窗口左边以全屏显示，这时【Full Size】按钮变成【End full size】按钮，单击此按钮可以返回先前的状态；单击【Reconstruct】按钮，可以将该图像系数进行重构，并显示重构后的图像。

5．树模式

在图 2-41 中选择 Tree 模式，出现图 2-42 所示结果。

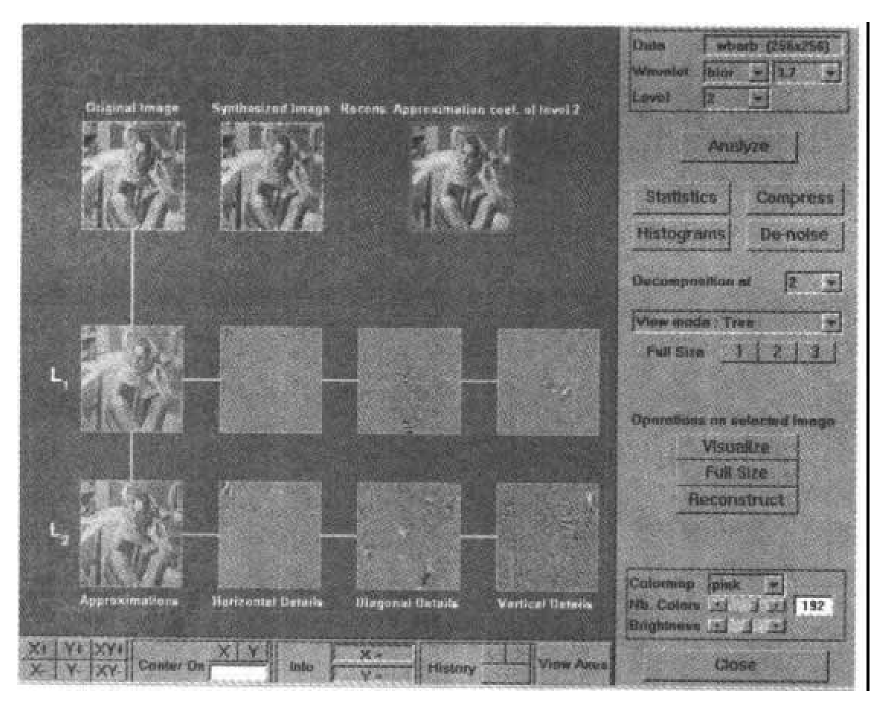


图 2-42　Tree 模式

6．放大细节

拖动鼠标选择想放大的图像部分，然后单击图 2-42 中右下方的【XY+】按钮，即同时进行水平和垂直方向上的放大，如图 2-43 所示。要想回到原来的状态，只需单击图下方的 History【<<—】按钮。

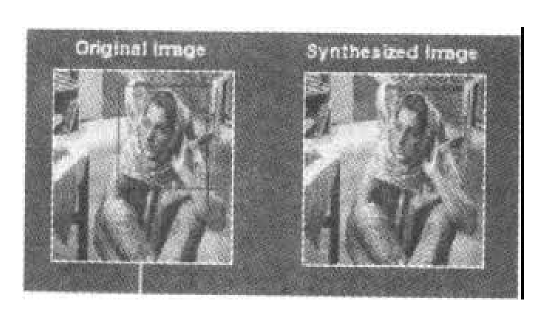




图 2-43　放大细节

7．图像压缩

单击图 2-42 右上方的【Compress】按钮，出现图 2-44 所示的图像压缩窗口。

二维离散小波分析图像工具箱可以自动选择阈值水平，以在保存图像信号的能量和减少压缩和保留的系数之间折中。当然，用户也可在 Select Global Threshold 选择框中拖动滚动条或在右边的文本框中输入数值进行手动设置。由于这里是通过对图像小波分解后的高频系数进行阈值量化来进行数据压缩，而高频系数有水平方向、对角线方向和垂直方向三个部分，所以图中也相应有三个选项可选。用户可以分别设置阈值，再单击【Compress】按钮进行图像数据的压缩。

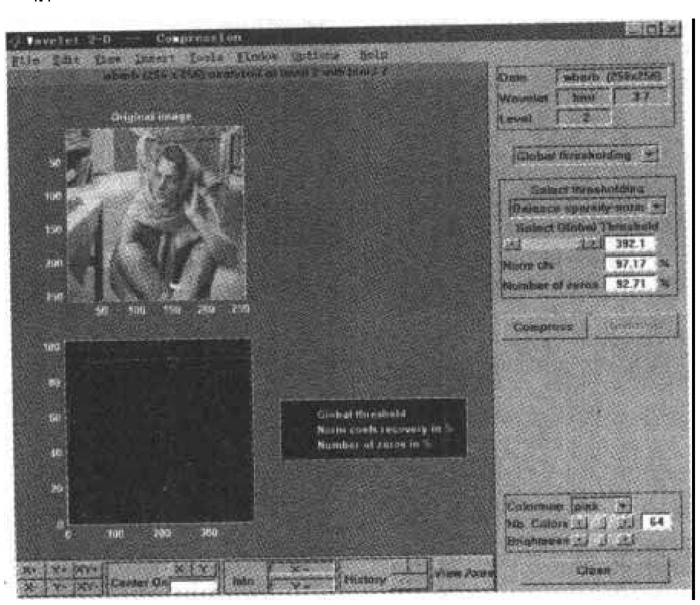


图 2-44　图像压缩窗口

下面我们选择 By Level thresholding，在 Select thresholding 中选择 Remove near 0，然后单击【Compress】按钮，则原始图像和压缩后的图像并列显示如图 2-45 所示。可以看出，尽管压缩去掉了将近一半的系数，图像的清晰度没有明显的降低。

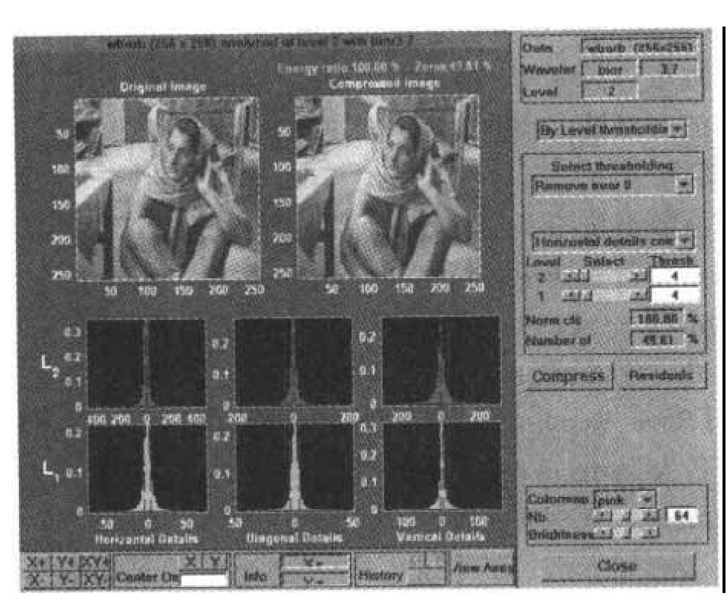


图 2-45　原始图像和压缩后的图像

8．显示其他信息

在 Wavelet 2-D Compress 工具中单击【Residuals】按钮，出现 More on Residuals for Wavelet 1-D Copression 窗口，如图 2-46 所示。

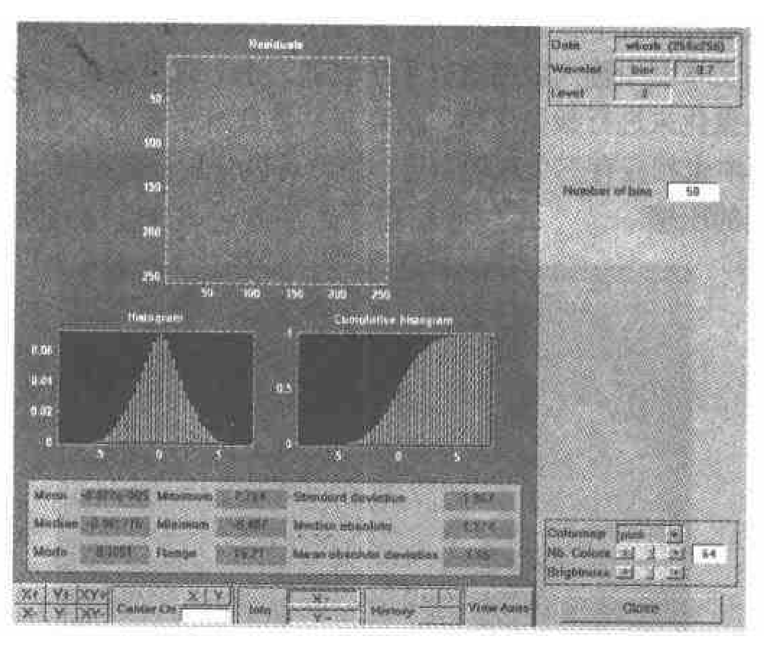


图 2-46　其他信息

可以看出，除均值、模式、中值、误差、标准差等统计量外，频域分布图（直方图和求和形式的直方图）也被显示。

2.4.3　图形接口方式中的信息交互

二维离散小波图形工具可以让用户从磁盘引入或输出信息，如图 2-40 所示的【File】下拉菜单。

1．从二维离散小波图形工具保存信息到磁盘

用户可以将二维离散小波图形工具产生的合成后的图像、系数和分解后的图像保存到磁盘，这些信息以后还可以被二维离散小波图形工具重新利用。

（1）保存合成图像

用户可以将二维离散小波图形工具处理后的信号以 MAT 格式的文件保存起来。例如，首先调入一个图形工具自带的演示分析信号，选择菜单命令【File】→【Demo Analysis】→【at level 3 with sym4】→【Detail Durer】，然后对其进行压缩。当关闭压缩或去噪窗口时，在出现的 update the synthesized signal 对话框中单击【Yes】按钮。再进入图形工具主菜单选择【File】→【Save Synthesized Image】命令，在出现的对话框中选择保存路径和文件名，这里我们取名为 symage.mat。为了将该信号装载进用户工作空间，只需要在命令提示符后键入 load symage，其保存的结构如下：

```
>>whos
Name        Size        Bytes       Class
X           359×371     1065512     double array
```

```
map         64×3        1536        double array
valTHR      1×1         8           double array
wname       1×4         8           char array
```

说明：这里 X 是合成图像信号，变量 map 包含了 colormap 信息，变量 valTHR 包含了全局阈值，wname 是压缩或消噪用的基本小波。

（2）保存离散小波变换系数

二维离散小波分析图形工具可让用户将 DWT 变换后的系数保存到磁盘。工具箱以用户选择的文件名将其保存在当前目录。还是以其自带的 Demo 为例：

选择菜单命令【File】→【Demo Analysis】→【at level 3 with sym4】→【Detail Durer】。

当以 cfsdurer.mat 保存小波变换系数后，再将其装载进工作空间：

```
>>load cfsdurer
>>whos
Name        Size          Bytes       Class
coefs       1×142299      1138392     double array
map         64×3          1536        double array
sizes       5×2           80          double array
valTHR      0×0           0           double array
wname       1×4           8           char array
```

说明：这里变量 coefs 包含了离散小波变换系数，sizes 是相关的矩阵大小。更精确地讲，在上面的例子里 coefs 是一个 1×142299 的向量，sizes 包含了系数的每一部分的长度。

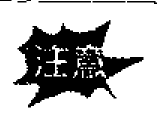
（3）保存分解后的图像信号

一维离散小波分析可以在当前的目录下以 wa2 扩展名来保存分解后的信号。还是以其自带的 Demo 为例：

选择菜单命令【File】→【Demo Analysis】→【at level 3，with sym4】→【Detail Durer】，

为了保存这次的分析结果，在主菜单中选择菜单命令【File】→【Save Decomposition】，然后在出现的对话框中键入文件名 decdurer.wa2，将其装载进工作区：

```
>>load decdurer.wa2 -mat
>>whos
Name        Size          Bytes       Class
coefs       1×142299      1138392     double array
data_name   1×6           12          char array
map         64×3          1536        double array
sizes       5×2           80          double array
valTHR      0×0           0           double array
wave_name   1×4           8           char array
```



保存选项在二维离散小波分析图形工具中的消噪和压缩窗口依然有效，这样用户就不必回到二维离散小波分析图形工具的主窗口就可以很方便地直接在消噪或压缩窗口保存相应的信息。

2. 装载信息到二维离散小波分析图形工具

用户可以装入图像、小波分解系数数据或小波分解后的图像到图形工具中进行分析。

这些信息可能是先前从图形工具中保存到磁盘然后又在工作区进行过处理，或者是用户通过命令行方式产生的。不管是何种情形，用户必须严格遵守二维小波工具所要求的文件格式和数据结构，否则就会产生出错信息。

（1）装载图像

小波工具箱只支持索引图像（Index Image），即相应的图像矩阵只包含从 1 到 n 的整数（n 为图像中的颜色数目）。例如，假定用户定义了一个名为 brain 的图像信号以利用 Wavelet 2-D 工具进行分析，首先执行以下命令：

```
>>X = brain;
>>map = pink(256);
>>save myfile X map
```

然后就可以通过菜单命令【File】→【Load Signal】装入该信号了。



图形工具箱允许用户装载不全是从 1 到 n 的整数的图像数据，但显示图像时将出现误差（对计算没有影响）。因为显示时小于 1 的值将被近似为 1，大于 n 的值被近似为 n，在 1 到 n 之间的非整数取最相近的整数

（2）装载二维离散小波变换系数

在装载之前，首先要有包含至少 coefs 和 sizes 两个变量的 MAT 格式文件。其格式和含义请参见前面的说明。图 2-47 是某个具体信号的示意图。

当在工作区编辑或构造好合适的信号后，键入：

```
>>save myfile coefs sizes
```

然后在菜单命令【File】→【Load Coefficients】中选择 myfile 即可。

（3）装载二维离散小波分解后的信号

在装载之前，要以 wa2 为扩展名将有关数据保存为 MAT 格式的文件，该数据文件必须包含 coefs、wave_name 和 sizes 三个变量，变量 map、data_name 是可选的。

当在工作区编辑或构造好合适的信号后，键入：

```
>>save myfile.wa2 coefs sizes wave_name
```

然后再单击菜单【File】→【Load Decomposition】将 myfile 装入。

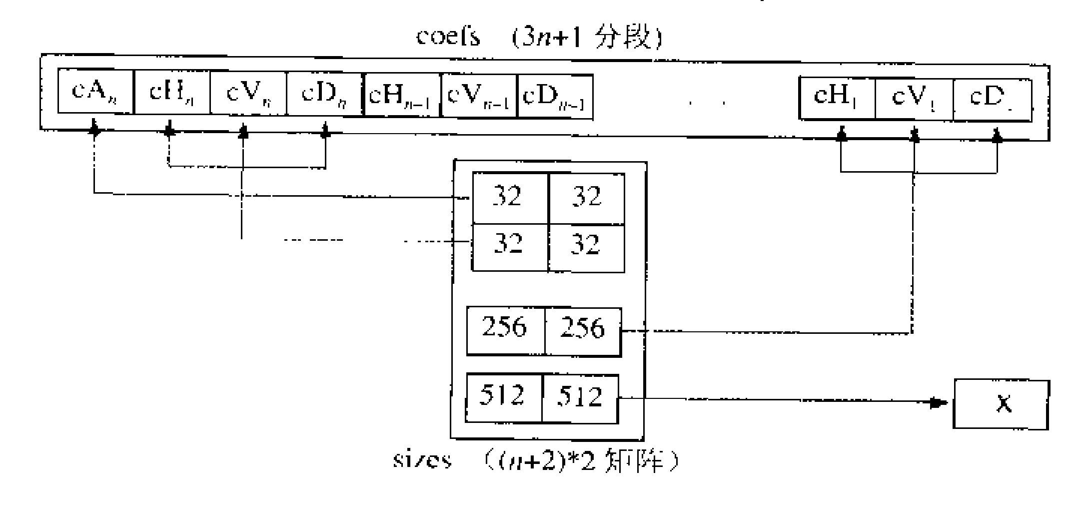


图 2-47 某具体信号的示意图

2.5 一维离散平稳小波分析

本节将介绍如何利用小波工具箱进行一维离散平稳小波分析的特征，涉及到的小波工具箱函数只有分解函数 swt 和合成函数 iswt，具体用法将在使用时讲解。

2.5.1 一维离散平稳小波分析——命令行方式

这里以一个含有噪声的多普勒测试信号为例，如图 2-48 所示。

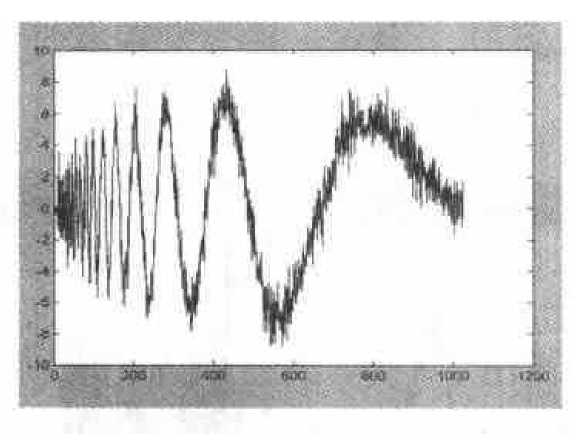


图 2-48 多普勒测试信号

1. 装载信号

在 MATLAB 命令窗口中输入：

```
>>load noisdopp;
>>s = noisdopp;
```

这里要说明的是，对于平稳小波变换 SWT，如果需要进行第 k 层次的分解，则 2^k 必须被信号长度整除。否则，用户必须利用信号延拓图形工具（Signal Extension GUI）或延拓函数 wextend 对信号进行延拓以满足上述要求。

2. 完成信号的单尺度一维离散平稳小波分解

采用 db1 基本小波来分解信号：

```
[swa,swd] = swt(s,1,'db1');
```

这就产生了低频系数 swa 和高频系数 swd，它们和原始信号具有相同的长度：

```
>>whos
  Name        Size       Bytes    Class
  noisdopp    1×1024     8192     double array
  s           1×1024     8192     double array
  swa         1×1024     8192     double array
  swd         1×1024     8192     double array
```

这里对单尺度一维离散平稳小波变换函数 swt 做一介绍。其格式为：

① SWC = swt(X,N,'wname')

② SWC = swt(X,N,Lo_D,Hi_D)

③ [SWA,SWD] = swt(X,N,'wname')

④ [SWA,SWD] = swt(X,N,Lo_D,Hi_D)

说明：其中 X 为被分析的离散信号，wname 为分解所用到的小波函数，Lo_D、Hi_D 为分解滤波器。格式①、②中 SWC 返回变换后的系数向量 SWC(i,:)（1≤i ≤N)，包含了第 i 层的高频系数，SWC(N+1,:)包含了第 N 层的低频系数。格式③、④中 SWA 和 SWD 分别返回平稳小波变换系数，保存格式 SWA(i,:)(1≤i≤N)，包含了第 i 层低频系数，SWD(i,:)（1≤ i≤ N）包含了第 i 层高频系数。

3. 显示低频和高频部分

为了显示第一层的分解结果，键入：

```
subplot(1,2,1), plot(swa);
title('Approximation cfs')
subplot(1,2,2), plot(swd);
title('Detail cfs')
```

结果如图 2-49 所示。

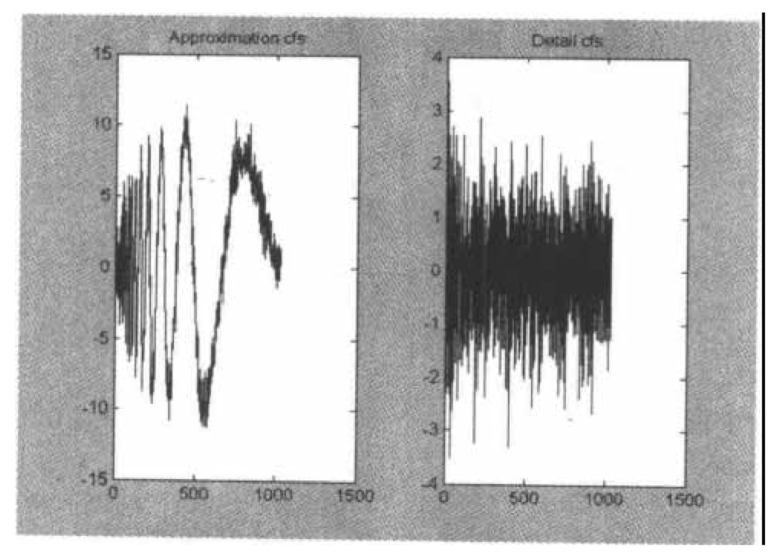


图 2-49 低频和高频部分

4. 由平稳小波逆变换重构信号

键入命令：

```
>>A0 = iswt(swa,swd,'db1');
```

查看重构误差：

```
>>err=norm(s-A0)
err =
    2.1450e-014
```

其中 iswt 为单尺度一维离散小波逆变换函数。 格式为：

① X = iswt(SWC, 'wname')

② X = iswt(SWA,SWD, 'wname')

③ X = iswt(SWC,Lo_R,Hi_R)

④ X = iswt(SWA,SWD,Lo_R,Hi_R)

说明：对格式①、③，它是用小波函数进行重构，对于格式②、④，是用重构滤波器进行重构。Lo_R 和 Hi_R 的长度是相等的。X 为重构后信号的向量。

5．从系数构建低频和高频部分

为了从系数 swa 和 swd 中构建第一层的低频和高频部分，键入：

```
nulcfs = zeros(size(swa));
A1 = iswt(swa,nulcfs,'db1');
D1 = iswt(nulcfs,swd,'db1');
```

6．显示低频和高频部分

为了显示上面的重构结果，键入：

```
subplot(1,2,1), plot(A1);
title('Approximation A1');
subplot(1,2,2), plot(D1);
title('Detail D1');
```

结果如图 2-50 所示。

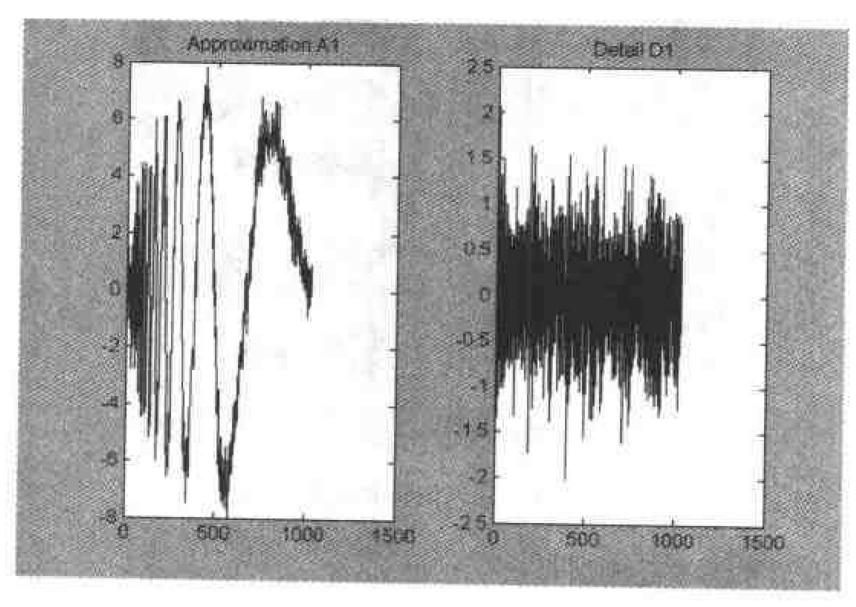


图 2-50　低频和高频部分

7．多层平稳小波分解

为了完成一个三尺度的分解（仍用 db1），键入：

```
>>[swa,swd] = swt(s,3,'db1');
```

观察 swa 和 swd 的结构：

```
>>clear A0 A1 D1 err nulcfs
>>whos
```

Name	Size	Bytes	Class
noisdopp	1×1024	8192	double array
s	1×1024	8192	double array
swa	3×1024	24576	double array
swd	3×1024	24576	double array

8．显示低频和高频系数

请参考例程 2-4。

例程 2-4

```
kp = 0;
for i = 1:3
subplot(3,2,kp+1), plot(swa(i,:));
title(['Approx. cfs level ',num2str(i)])
subplot(3,2,kp+2), plot(swd(i,:));
title(['Detail cfs level ',num2str(i)])
kp = kp + 2;
end
```

结果如图 2-51 所示。

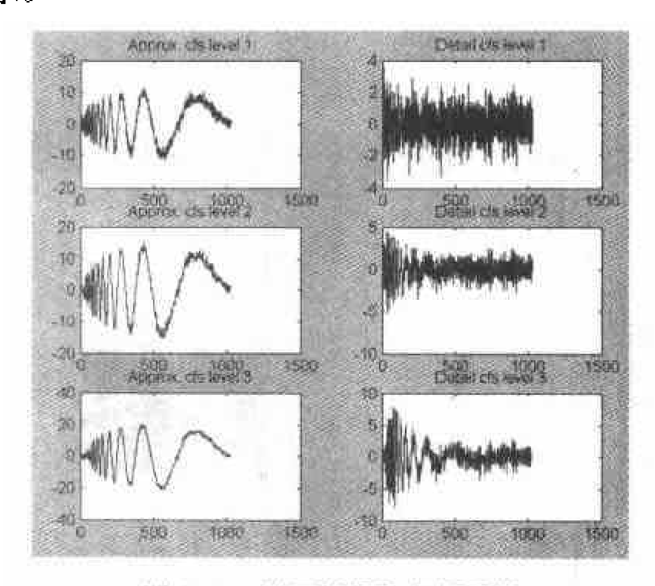


图 2-51　显示低频和高频系数

9．从系数重构第 3 层的低频信号

键入：

```
mzero = zeros(size(swd));
A = mzero;
A(3,:) = iswt(swa,mzero,'db1');
```

10．重构第 1、2、3 层的高频信号

键入：

```
D = mzero;
for i = 1:3
swcfs = mzero;
swcfs(i,:) = swd(i,:);
D(i,:) = iswt(mzero,swcfs,'db1');
end
```

11．从第 3 层的低频部分和第 2、3 层的高频部分重构第 1、2 层的低频部分

键入：

```
A(2,:) = A(3,:) + D(3,:);
A(1,:) = A(2,:) + D(2,:);
```

12．显示第 1、2、3 层的低频和高频部分

请参考例程 2-5。

例程 2-5

```
kp = 0;
for i = 1:3
subplot(3,2,kp+1), plot(A(i,:));
title(['Approx. level ',num2str(i)])
subplot(3,2,kp+2), plot(D(i,:));
title(['Detail level ',num2str(i)])
kp = kp + 2;
end
```

结果如图 2-52 所示。

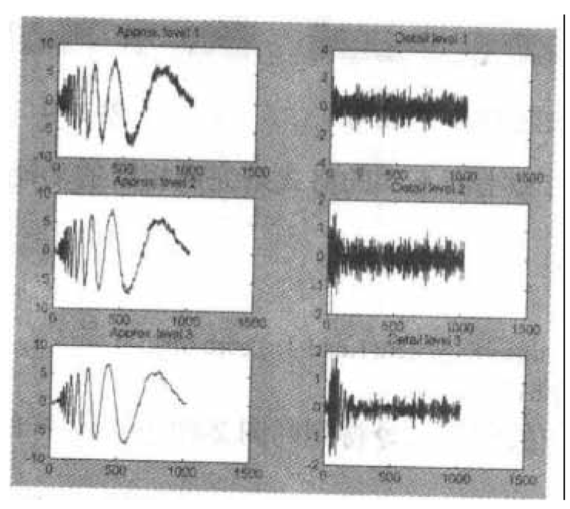


图 2-52　第 1、2、3 层的低频和高频部分

13．阈值去噪

为了去除信号中的噪声，我们用函数 ddencmp 来计算一个默认的全局阈值，用 wthresh 来对高频系数进行阈值处理，然后用函数 iswt 来获得消除噪声后的信号：

```
[thr,sorh] = ddencmp('den','wv',s);
dswd = wthresh(swd,sorh,thr);
clean = iswt(swa,dswd,'db1');
```

显示原始信号和去除了噪声的信号，如图 2-53 所示。

```
subplot(2,1,1), plot(s);
title('Original signal')
subplot(2,1,2), plot(clean);
title('De-noised signal')
```

从图 2-53 可以看出，去除噪声后的信号仍然含有一定的噪声。这可以通过将原来的三层次的分解改为五层次的分解并重复上面的步骤：

```
[swa,swd] = swt(s,5,'db1');
[thr,sorh] = ddencmp('den','wv',s);
dswd = wthresh(swd,sorh,thr);
```

```
clean = iswt(swa,dswd,'db1');
subplot(2,1,1), plot(s); title('Original signal')
subplot(2,1,2), plot(clean); title('De-noised signal')
```

显然，图 2-54 中的消噪效果比 2-53 要好。

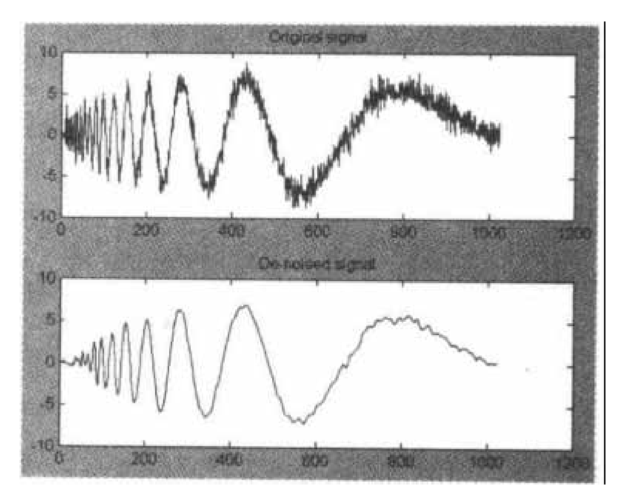


图 2-53　原始信号和去除了噪声的信号

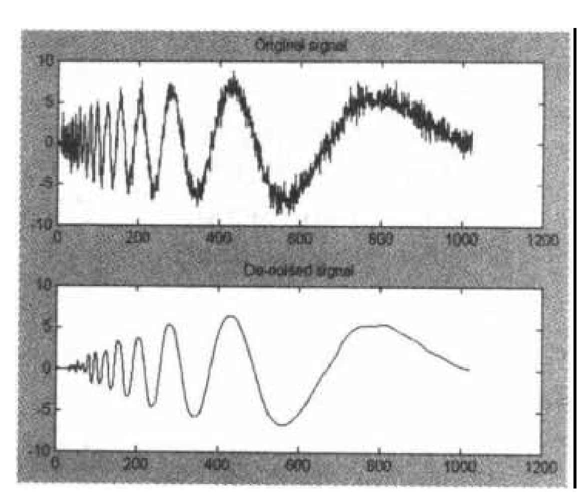


图 2-54　进一步消噪

对函数 swt 和 iswt 还有和上面不同的格式来完成相同的功能：

```
lev = 5;
swc = swt(s,lev,'db1');
swcden = swc;
swcden(1:end-1,:) = wthresh(swcden(1:end-1,:),sorh,thr);
clean = iswt(swcden,'db1');
```

然后可以用上面步骤中的作图命令得到同图 2-54 所示一样的结果。

2.5.2　一维离散平稳小波分析（消噪）——图形接口方式

1．启动一维离散平稳小波图形工具

在图 2-4 所示小波工具箱主菜单中选择 SWT De-noising 1-D，出现图 2-55 所示的一维离散平稳小波分析图形工具。

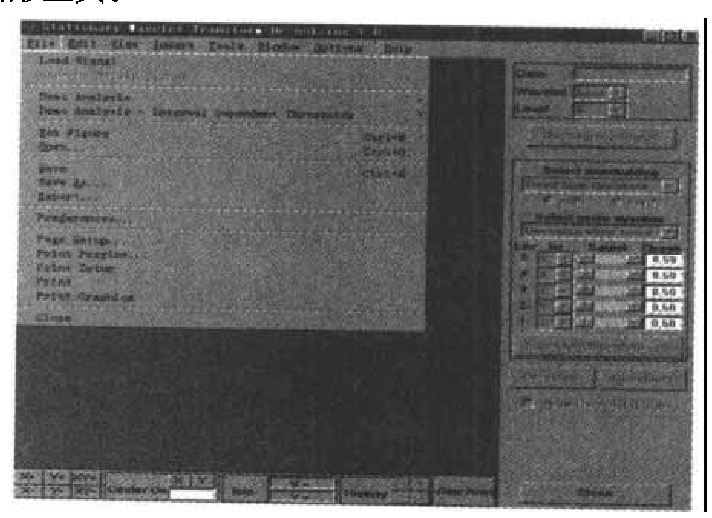


图 2-55　一维离散平稳小波分析图形工具

2．装载数据

在图 2-55 中单击【File】→【Load Signal】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noisbloc.mat 文件。

3．对信号进行一维平稳小波变换

在图 2-55 中的右上角选择基本小波为 db1 和尺度 Level 为 5。选择好以上参数后，就可以单击【Decompose Signal】按钮。经过短时间的计算后将出现如图 2-56 所示的分解结果。

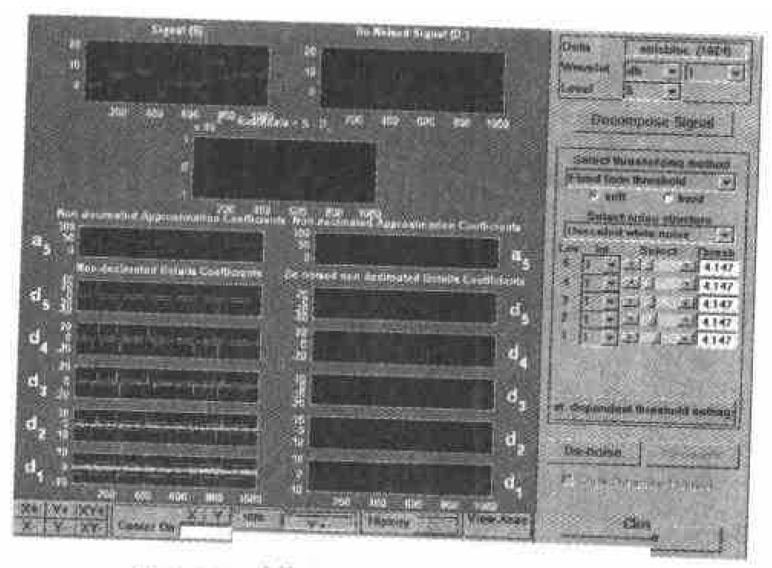


图 2-56　对信号进行一维平稳小波变换

4．利用平稳小波变换来消噪

我们选择固定软阈值（Fixed form soft thresholding）和尺度未知的白噪声（unscaled white noise）。窗口右边的滑动条用来指示和调整相应各层次的阈值，结果以黄点线反映在窗口左边各层次高频部分的图形中。用户也可直接用鼠标左键拖动黄点线来改变阈值。用户可注意到低频系数并没有进行阈值处理。

选择好阈值后，单击【De-noise】按钮，结果如图 2-57 所示。

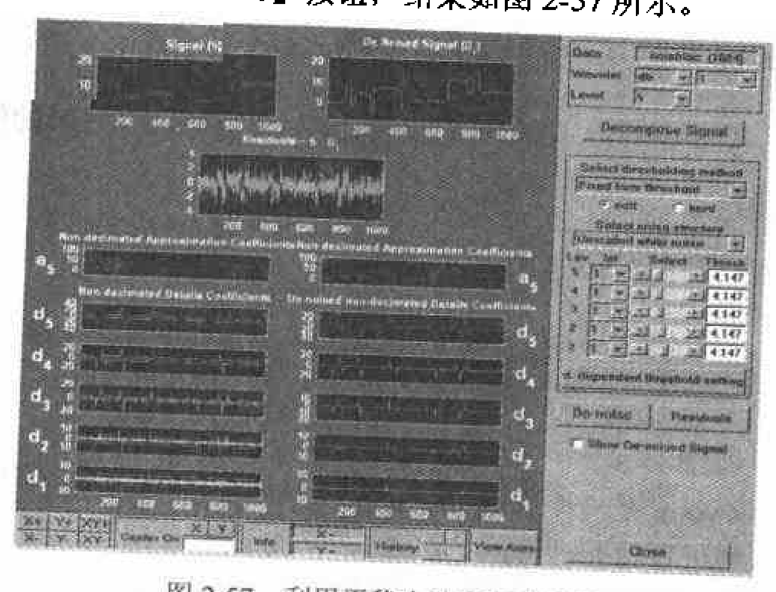


图 2-57　利用平稳小波变换来消噪



也可重新选择硬阈值（Fixed form hard thresholding）来进行消噪。具体的消噪处理请参见第 3 章，这里只是给读者一个感性认识。

2.5.3 图形接口方式中的信息交互

在进行完上面的消噪处理后，可以保存经过消噪处理后的信号。在图 2-57 的主菜单中选择【File】→【Save De-noised Signal】，以 dnoibloc.mat 保存。然后将其装载进工作空间：

```
>>load dnoibloc
>>whos
  Name          Size        Bytes      Class
  dnoibloc      1×1024      8192       double array
  thrParams     1×5         580        cell array
  wname         1×3         6          char array
```

说明：消噪的参数由小波基函数（wname）给出，而与层次相关的阈值信息保存在长度为 5 的 thrParams 变量里。

i 从 1 到 5，thrParams{i}包含了阈值间隔的上下限及阈值，例如对 level 1：

```
>>thrParams{1}
ans =
    1.0e+003 *
    0.0010    1.0240    0.0041
```

这里下限是 1，上限是 1024，阈值是 4.1。

2.6 二维离散平稳小波分析

本节将介绍如何利用小波工具箱进行二维离散平稳小波分析。本节涉及到的用于二维图像分析的小波工具箱函数只有 swt2 和 iswt2，其格式和用法将在下面讲解。

2.6.1 二维离散平稳小波分析——命令行方式

本节中我们举一个应用二维小波平稳分析来对一个图像进行消噪的例子。

1．装载图像信号

在 MATLAB 命令行窗口中输入：

```
>>load noiswom;
>>whos
  Name      Size       Bytes      Class
  X         96×96      73728      double array
  map       255×3      6120       double array
```

说明：这里 noiswom 是一个含有噪声的女人图像。值得一提的是，对于平稳小波变换（SWT），如果需要进行第 k 层分解，则 2^k 必须被 size（X，1）和 size（X，2）整除。如果原始信号不满足这个要求，则应该先用图像延拓图形工具（Image Extension GUI）或

函数 wextend 来对原始图像进行延拓以使其满足上述要求。

2．完成图像的单层次小波分解

这里采用 db1 基小波：

```
[swa,swh,swv,swd] = swt2(X,1,'db1');
```

这就计算出了第一层的低频系数 swa，以及高频系数 swh（水平）、swv（垂直）、swd（对角线）。

这里 swt2 是二维离散平稳小波变换函数，其格式为：

① SWC = swt2(X,N,'wname')
② [A,H,V,D] = swt2(X,N,'wname')
③ SWC = swt2(X,N,Lo_D,Hi_D)
④ [A,H,V,D] = swt2(X,N,Lo_D,Hi_D)

说明：格式①、②利用基小波 wname 计算 X 在 N 层的平稳小波变换，N 是严格的正整数，2^N 必须被 size（X，1）和 size（X，2）整除。输出[A,H,V,D]是三维的数组，包含了相应的系数：

对 1≤ i≤ N，矩阵 A(:,:,i)包含了第 i 层的低频系数；

矩阵 H(:,:,i)、V(:,:,i)和 D(:,:,i)包含了第 i 层的高频系数（水平、垂直和对角线方向）：

SWC = [H(:,:,1:N) ; V(:,:,1:N) ; D(:,:,1:N) ; A(:,:,N)]

格式③、④和①、②的区别只是在于是利用分解滤波器进行平稳小波变换，其他的含义都相同。

可以观察各个系数向量的结构：

```
>>whos
  Name      Size       Bytes      Class
  X         96×96      73728      double array
  map       255×3      6120       double array
  swa       96×96      73728      double array
  swd       96×96      73728      double array
  swh       96×96      73728      double array
  swv       96×96      73728      double array
```

3．显示低频和高频系数

例程 2-6 可以显示第一层的低频和高频系数。

例程 2-6

```
map = pink(size(map,1));
colormap(map)
subplot(2,2,1), image(wcodemat(swa,192));
title('Approximation swa')
subplot(2,2,2), image(wcodemat(swh,192));
title('Horiz. Detail swh')
subplot(2,2,3), image(wcodemat(swv,192));
title('Vertical Detail swv')
subplot(2,2,4), image(wcodemat(swd,192));
title('Diag. Detail swd')
```

结果如图 2-58 所示。

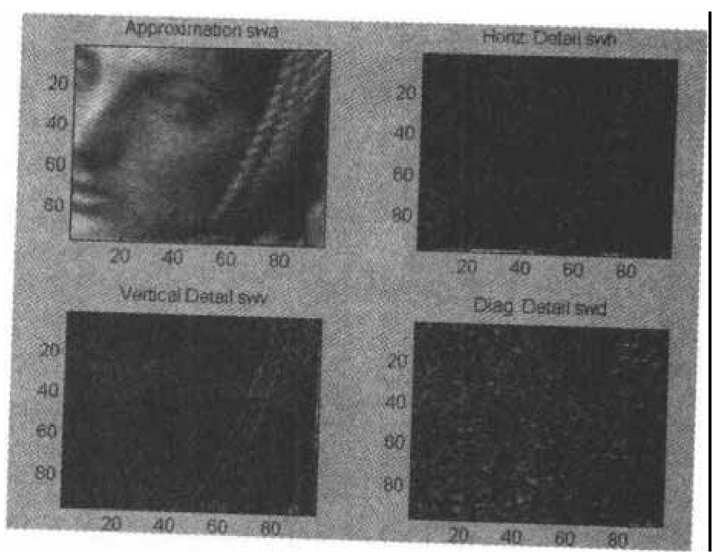


图 2-58 显示低频和高频系数

4. 通过平稳小波逆变换重构图像

在命令窗口键入：

```
>>A0 = iswt2(swa,swh,swv,swd,'db1');
```

检查重构误差：

```
>>err = max(max(abs(X-A0)))
err =1.1369e-013
```

这里 iswt2 是多尺度二维平稳小波重构函数，其格式为：

① X = iswt2(SWC,'wname')
② X = iswt2(A,H,V,D,'wname')
③ X = iswt2(SWC,Lo_R,Hi_R)
④ X = iswt2(A,H,V,D,Lo_R,Hi_R)

5. 从系数中重构第一层的低频和高频部分

从第三步中产生的系数 swa、swh、swv 和 swd 构造第一层的低频和高频（A1、H1、V1 和 D1）部分：

```
nulcfs = zeros(size(swa));
A1 = iswt2(swa,nulcfs,nulcfs,nulcfs,'db1');
H1 = iswt2(nulcfs,swh,nulcfs,nulcfs,'db1');
V1 = iswt2(nulcfs,nulcfs,swv,nulcfs,'db1');
D1 = iswt2(nulcfs,nulcfs,nulcfs,swd,'db1');
```

6. 显示低频和高频部分

例程 2-7 是为了显示第一层的分解结果。

例程 2-7

```
colormap(map)
subplot(2,2,1), image(wcodemat(A1,192));
title('Approximation A1')
```

```
subplot(2,2,2), image(wcodemat(H1,192));
title('Horiz. Detail H1')
subplot(2,2,3), image(wcodemat(V1,192));
title('Vertical Detail V1')
subplot(2,2,4), image(wcodemat(D1,192));
title('Diag. Detail D1')
```

结果如图 2-59 所示。

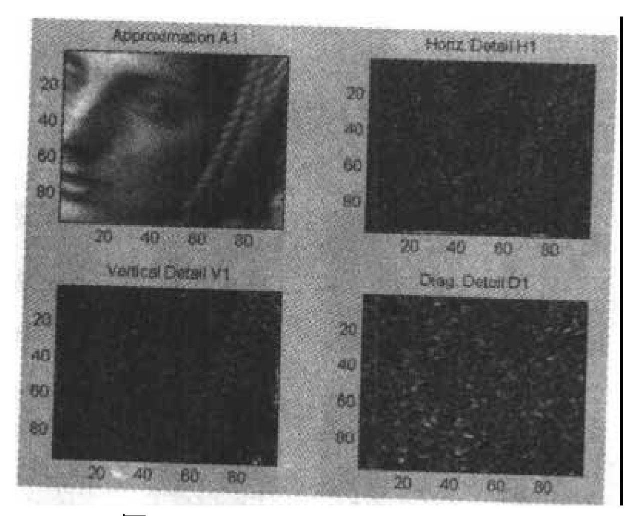


图 2-59 显示低频和高频部分

7．图像的多尺度二维离散平稳小波分解

仍然采用 db1 来完成多尺度二维离散平稳小波分解：

```
[swa,swh,swv,swd] = swt2(X,3,'db1');
```

这里计算返回的 swa 是第 1、2、3 层的低频系数，swh、swv 和 swd 是高频系数。可观察它们的存储结构：

```
>>clear A0 A1 D1 H1 V1 err nulcfs
>>whos
  Name      Size          Bytes     Class
  X         96×96         73728     double array
  map       255×3         6120      double array
  swa       96×96×3       221184    double array
  swd       96×96×3       221184    double array
  swh       96×96×3       221184    double array
  swv       96×96×3       221184    double array
```

8．显示多尺度二维平稳小波分解结果

请参考例程 2-8。

例程 2-8

```
colormap(map)
kp = 0;
for i = 1:3
subplot(3,4,kp+1), image(wcodemat(swa(:,:,i),192));
```

```
title(['Approx. cfs level ',num2str(i)])
subplot(3,4,kp+2), image(wcodemat(swh(:,:,i),192));
title(['Horiz. Det. cfs level ',num2str(i)])
subplot(3,4,kp+3), image(wcodemat(swv(:,:,i),192));
title(['Vert. Det. cfs level ',num2str(i)])
subplot(3,4,kp+4), image(wcodemat(swd(:,:,i),192));
title(['Diag. Det. cfs level ',num2str(i)])
kp = kp + 4;
end
```

结果如图 2-60 所示。

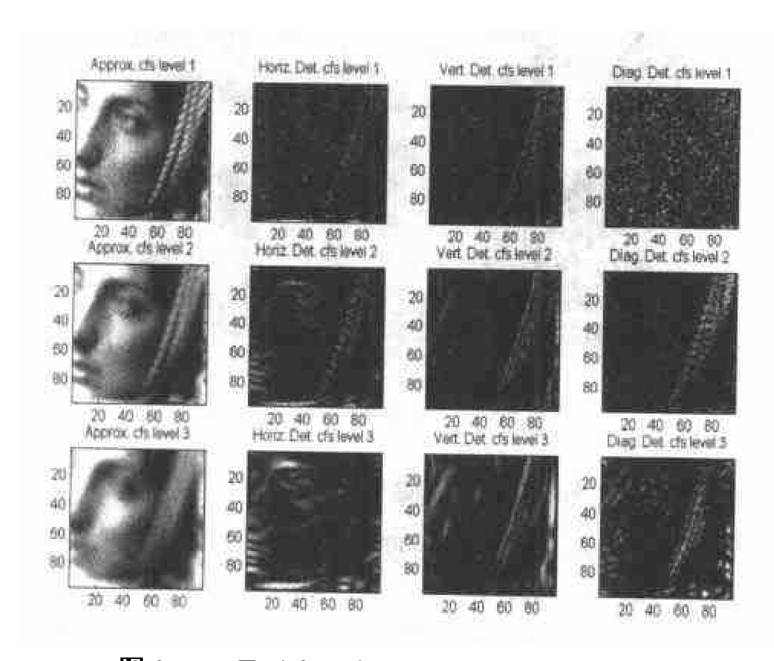


图 2-60　显示多尺度二维平稳小波分解结果

9. 从系数中重构第 3 层的低频信号

键入：

```
mzero = zeros(size(swd));
A = mzero;
A(:,:,3) = iswt2(swa,mzero,mzero,mzero,'db1');
```

10. 从系数中重构高频信号

例程 2-9 是为了重构第 1、2、3 层的高频信号。

例程 2-9

```
H = mzero; V = mzero;
D = mzero;
for i = 1:3
swcfs = mzero; swcfs(:,:,i) = swh(:,:,i);
H(:,:,i) = iswt2(mzero,swcfs,mzero,mzero,'db1');
swcfs = mzero; swcfs(:,:,i) = swv(:,:,i);
V(:,:,i) = iswt2(mzero,mzero,swcfs,mzero,'db1');
```

```
swcfs = mzero; swcfs(:,:,i) = swd(:,:,i);
D(:,:,i) = iswt2(mzero,mzero,mzero,swcfs,'db1');
End
```

11．重构第 1、2 层的低频部分

下面的命令通过第 3 层的低频部分和第 1、2、3 层的高频部分重构第 1、2 层的低频部分：

```
A(:,:,2) = A(:,:,3) + H(:,:,3) + V(:,:,3) + D(:,:,3);
A(:,:,1) = A(:,:,2) + H(:,:,2) + V(:,:,2) + D(:,:,2);
```

12．显示第 1、2、3 层的低频和高频部分

请参考例程 2-10。

例程 2-10

```
colormap(map)
kp = 0;
for i = 1:3
subplot(3,4,kp+1), image(wcodemat(A(:,:,i),192));
title(['Approx. level ',num2str(i)])
subplot(3,4,kp+2), image(wcodemat(H(:,:,i),192));
title(['Horiz. Det. level ',num2str(i)])
subplot(3,4,kp+3), image(wcodemat(V(:,:,i),192));
title(['Vert. Det. level ',num2str(i)])
subplot(3,4,kp+4), image(wcodemat(D(:,:,i),192));
title(['Diag. Det. level ',num2str(i)])
kp = kp + 4;
end
```

运行结果如图 2-61 所示。

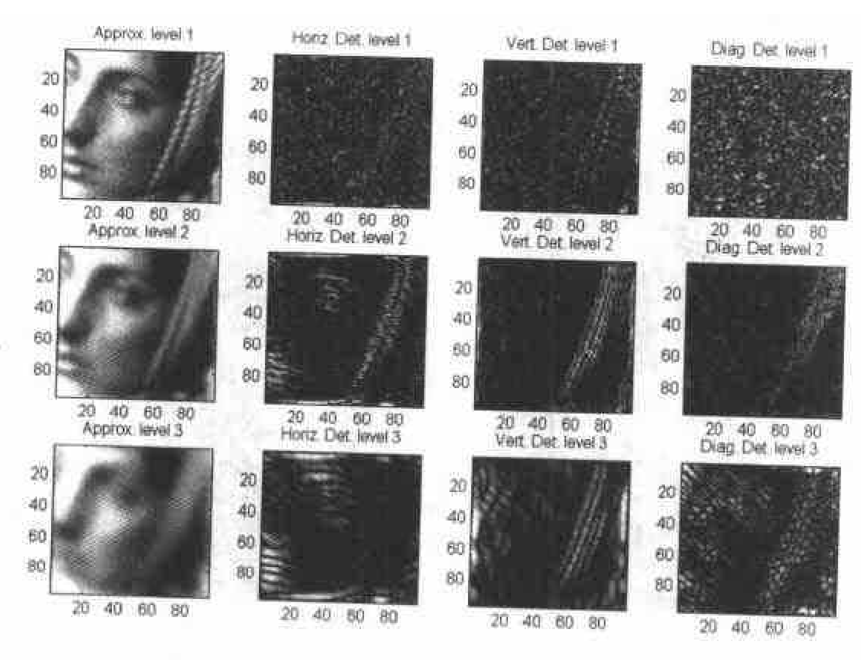


图 2-61　第 1、2、3 层的低频和高频部分

13．阈值消噪

为了对图像消噪，首先利用二维离散平稳小波分析图形工具 GUI tool（参见 2.6.2 小节）找到阈值 thr，然后利用 wthresh 函数对高频系数进行实际的阈值处理，最后利用函数 iswt2 来获得消噪后的图像。详见例程 2-11。

例程 2-11

```
thr = 44.5;
sorh = 's';
dswh = wthresh(swh,sorh,thr);
dswv = wthresh(swv,sorh,thr);
dswd = wthresh(swd,sorh,thr);
clean = iswt2(swa,dswh,dswv,dswd,'db1');
```

14．对比消噪前后的图像

请参考例程 2-12。

例程 2-12

```
thr = 44.5;
sorh = 's';
dswh = wthresh(swh,sorh,thr);
dswv = wthresh(swv,sorh,thr);
dswd = wthresh(swd,sorh,thr);
clean = iswt2(swa,dswh,dswv,dswd,'db1');
colormap(map)
subplot(1,2,1), image(wcodemat(X,192));
title('Original image')
subplot(1,2,2), image(wcodemat(clean,192));
title('De-noised image')
```

运行结果如图 2-62 所示。

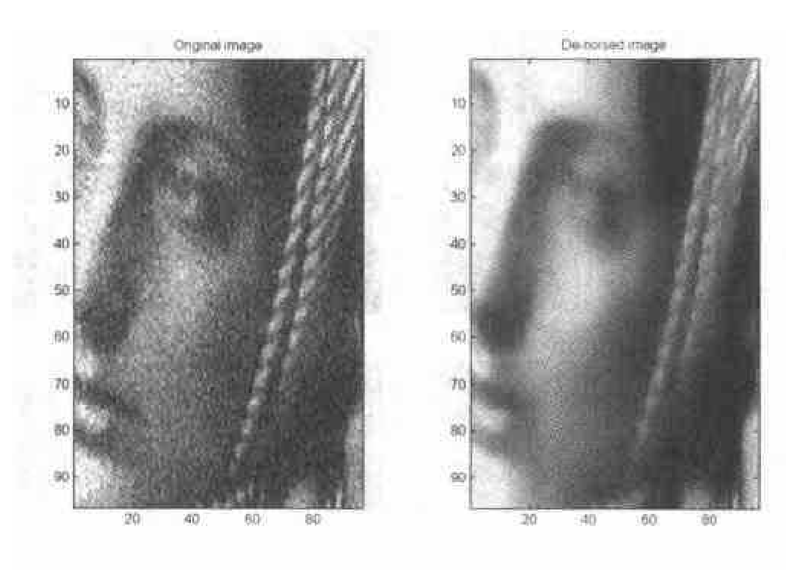


图 2-62 消噪前后的图像



完成上面的功能也可以采用下面的命令：

```
lev = 4;
swc = swt2(X,lev,'db1');
swcden = swc;
swcden(:,:,1:end-1) = wthresh(swcden(:,:,1:end-1),sorh,thr);
clean = iswt2(swcden,'db1');
```

然后在运行前面的命令后可得到同图 2-62 一样的结果。

2.6.2　二维离散平稳小波分析（消噪）——图形接口方式

下面我们采用 MATLAB 6.5 的小波分析图形工具箱来完成上面的分析。

1．启动二维离散平稳小波消噪图形工具

在图 2-4 中的小波工具箱主菜单中选择 SWT De-noising 2-D，出现图 2-63 所示的二维离散小波平稳小波消噪图形工具。

2．装载图像数据

在图 2-63 中单击【File】→【Load Image】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noiswom.mat 文件。

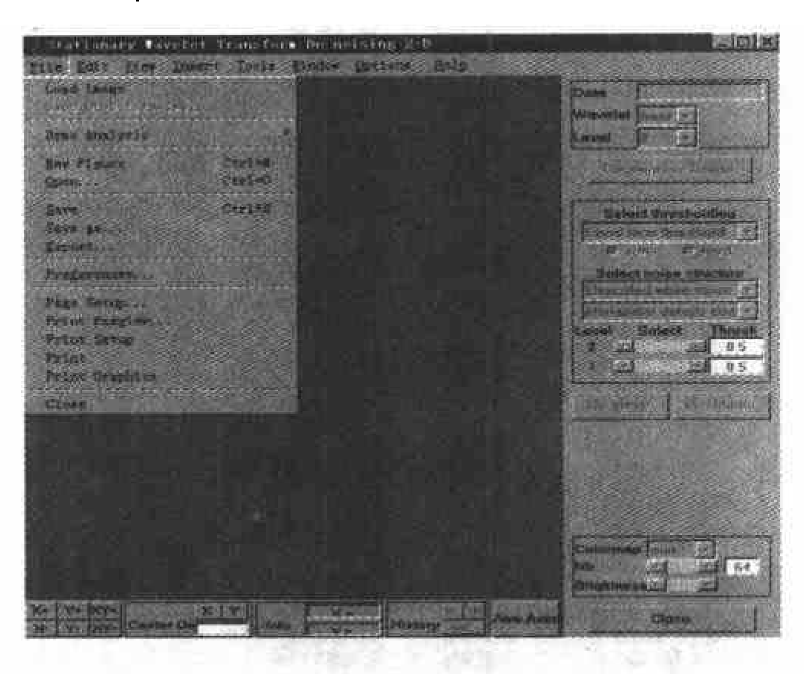


图 2-63　二维离散小波平稳小波消噪图形工具

3．对图像进行二维离散平稳小波分解

在图 2-63 的右上角选择基本小波为 harr 和尺度数 Level 为 4。选择好以上参数后，就可以单击【Decompose Image】按钮，则经过短时间的计算后将出现如图 2-64 所示的分解结果。

可以看出，图像分解后的小波系数的柱状图在窗口左边显示出来，从下向上依次是第 1、2、3、4 层，从左到右依次是高频部分的水平、对角线和垂直系数。

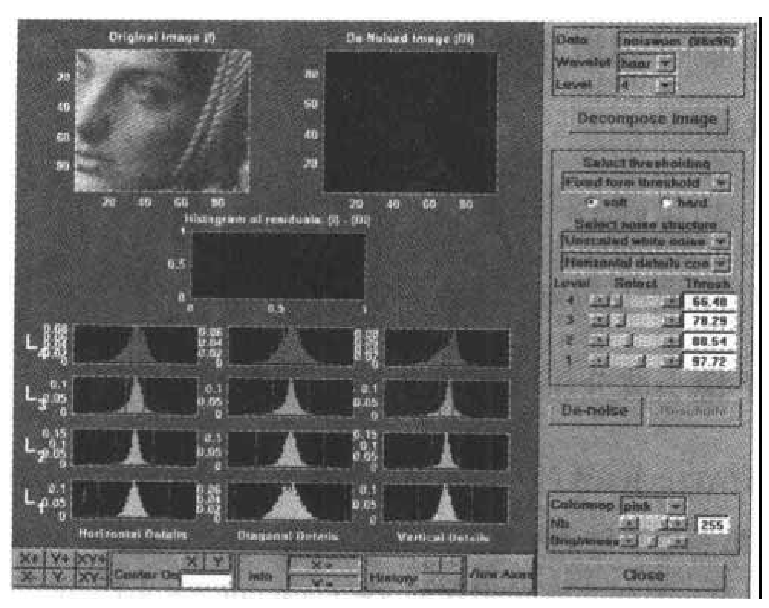


图 2-64　二维离散平稳小波分解

4．利用平稳小波变换对图像消噪

消噪算法有多种选择，这里我们仍然选择固定软阈值（fixed form soft thresholding）和尺度未知的白噪声（unscaled white noise）。窗口右边的滑动条用来指示和调整相应各层次的阈值，结果以黄点线反映在窗口左边各层次高频部分的柱状图中。选择好阈值后，单击【De-noise】按钮，结果如图 2-65 所示。

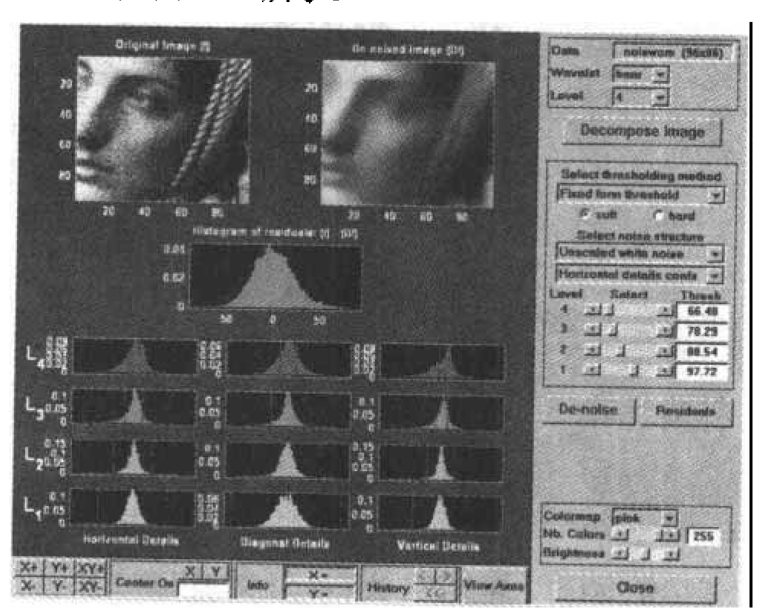


图 2-65　利用平稳小波变换对图像消噪

从图 2-65 可以看出，中间标题为 Histogram of residuals 图形对应的是原始图像中引入的噪声信号，近似于高斯分布。

5．选择阈值类型

单击图 2-65 中的 Select thresholding method 下拉框，选择阈值类型为 Penalize low，下面相关的阈值模式自动改为硬阈值（hard）。调整下面的 Sparsity 滑动条使阈值为 44.5，

然后单击【De-noise】按钮。

结果如图 2-66 所示。可以看出，消噪好的图像效果很好。

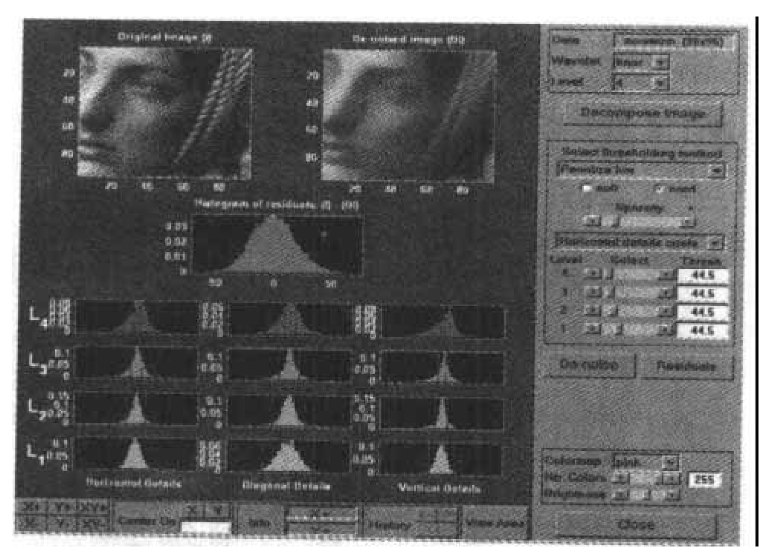


图 2-66　选择阈值类型

2.6.3　图形接口方式中的信息交互

在进行完上面的消噪处理后，可以保存经过消噪处理后的信号。在图 2-66 的主菜单中选择【File】→【Save De-noised Image】，以 dnoiwom.mat 保存。然后将其装载进工作空间：

```
>>load dnoiswom
>>whos
```

Name	Size	Bytes	Class
X	96×96	73728	double array
map	255×3	6120	double array
valTHR	3×4	96	double array
wname	1×4	8	char array

说明：X 是消噪后的图像，valTHR 保存了与层次相关的阈值。ValTHR 是一个 3×4 的矩阵，行对应的是三个方向（水平、垂直、对角线），列对应的是分解的四个层次。

2.7　一维小波回归估计

这一节将介绍如何利用 MATLAB 6.5 小波分析工具箱中的专有工具进行一维小波回归估计。该专有工具是图形工具，可以针对等间隔或非等间隔采样的数据进行消噪策略上的探究。

在进行下面的分析之前，首先通过下面的命令将延拓模式切换到“对称填充”（symmetric padding）：

```
dwtmode('sym');
```

具体含义可参见后面有关章节。

2.7.1 一维等间距观测估计（确定性设计 Fixed Design）

1. 启动一维回归估计图形工具

在图 2-4 中的小波工具箱主菜单中选择 Regression Estimation 1-D，出现图 2-67 所示的一维回归估计图形工具。

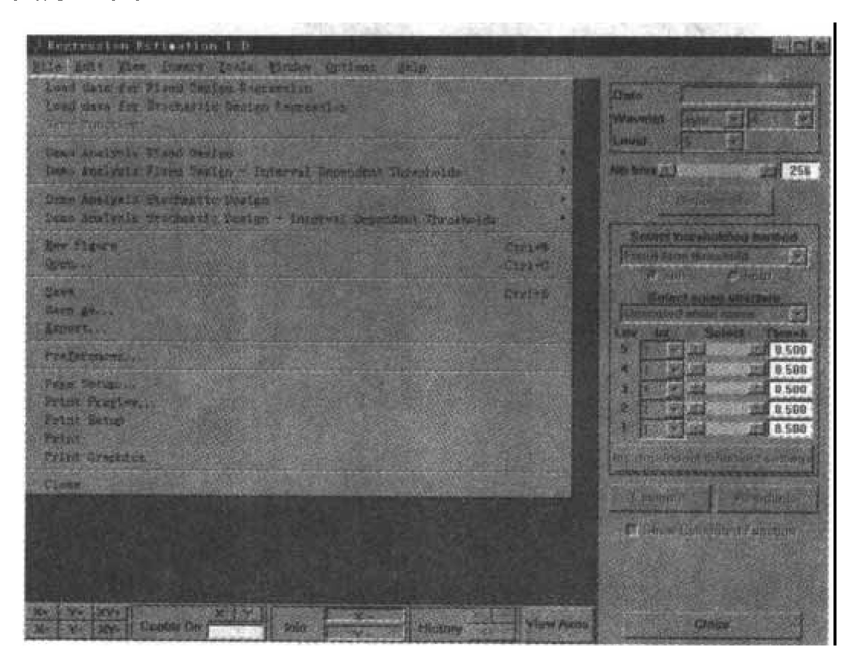


图 2-67　一维回归估计图形工具

在图 2-67 中的【File】菜单里选择【Load Data for Fixed Design Regression】命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noisbloc.mat 文件。结果如图 2-68 所示。

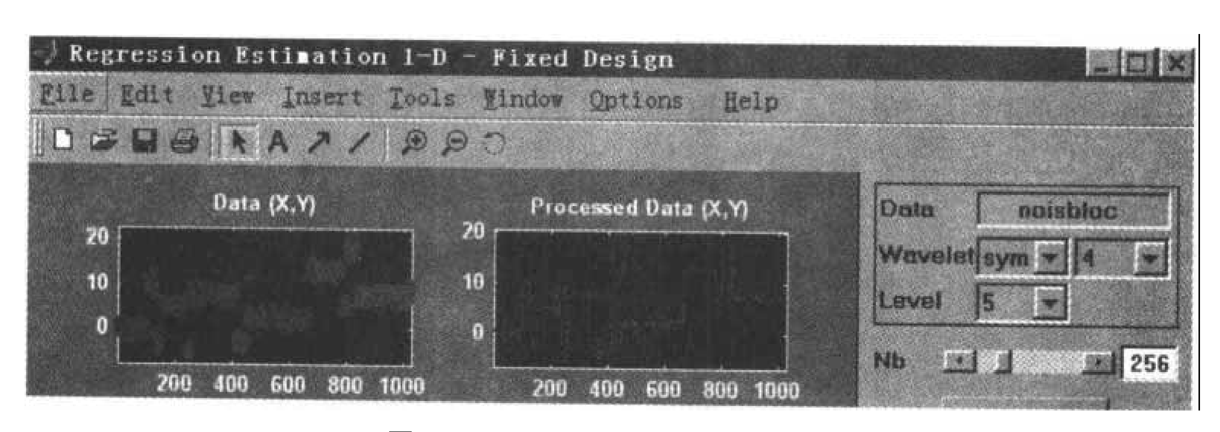


图 2-68　noisbloc.mat 文件运行结果

2. 选择处理后的数据

处理初始信号默认的 bins 值是 256，我们可以改变该值。在 Nb bins 编辑框中输入 64 或拖动滑动条使其为 64 并回车，观察图 2-69 与图 2-68 的区别。

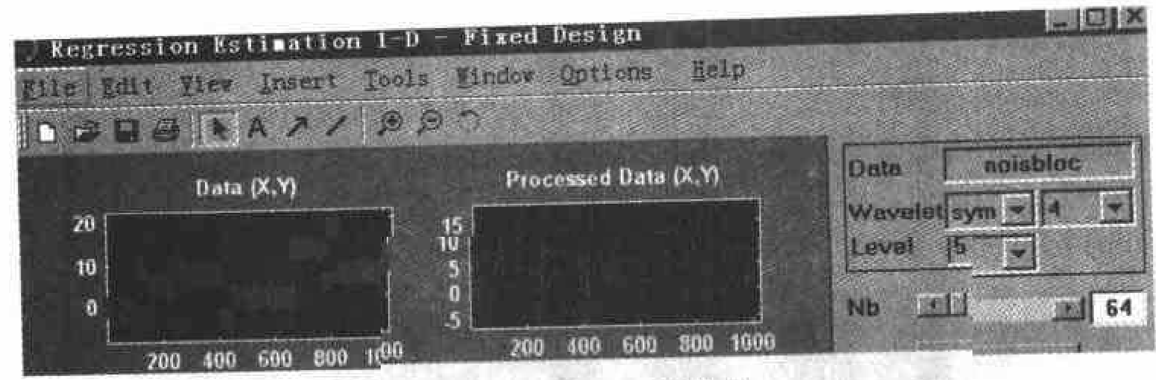


图 2-69　选择处理后的数据 1

然后再取 1000，结果如图 2-70 所示。

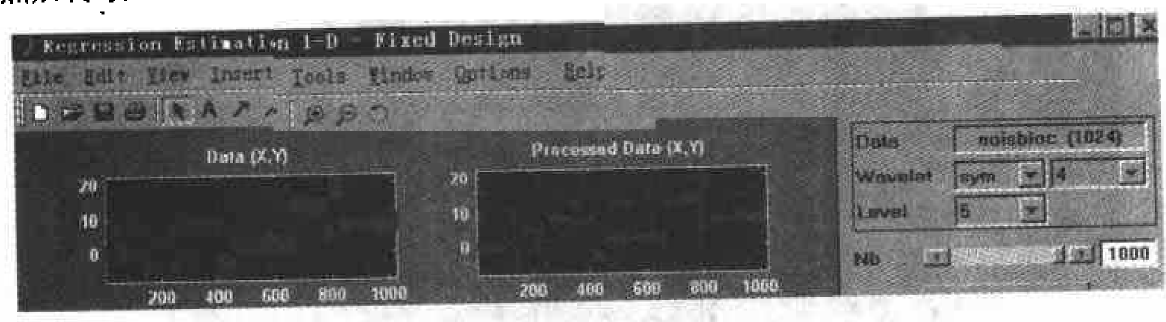


图 2-70　选择处理后的数据 2

可以看出，图 2-70 中右边的信号和初始信号非常接近，因为初始信号 noisbloc 的长度本身就是 1024。

3．对处理后的信号进行小波分解

在图 2-70 中的右上角选择基本小波为 harr 系和尺度数 Level 为 5。选择好以上参数后，就可以单击【Decompose】按钮，则经过短时间的计算后将出现如图 2-71 所示的分解结果。

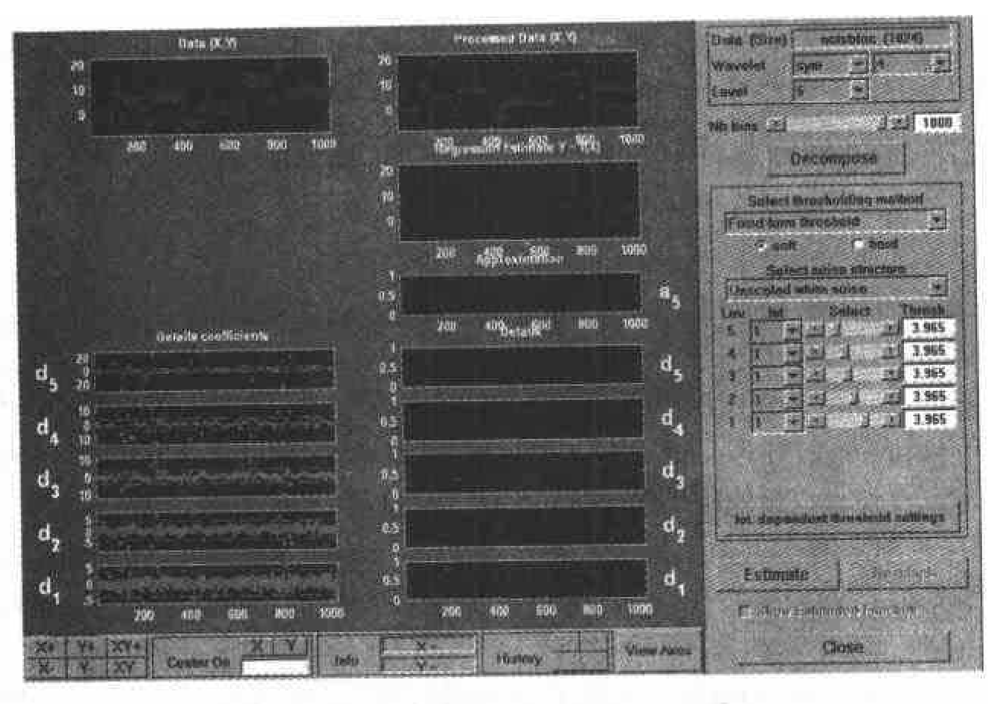


图 2-71　小波分解

4. 回归估计

尽管估计算法有多种选择，这里我们仍然选择系统默认的固定软阈值（Fixed form soft thresholding）和尺度未知的白噪声（unscaled white noise）。窗口右边的滑动条用来指示和调整相应各层次的阈值，结果以黄点线反映在窗口左边各层次高频部分的图形中。然后单击【Estimate】按钮，结果如图 2-72 所示。

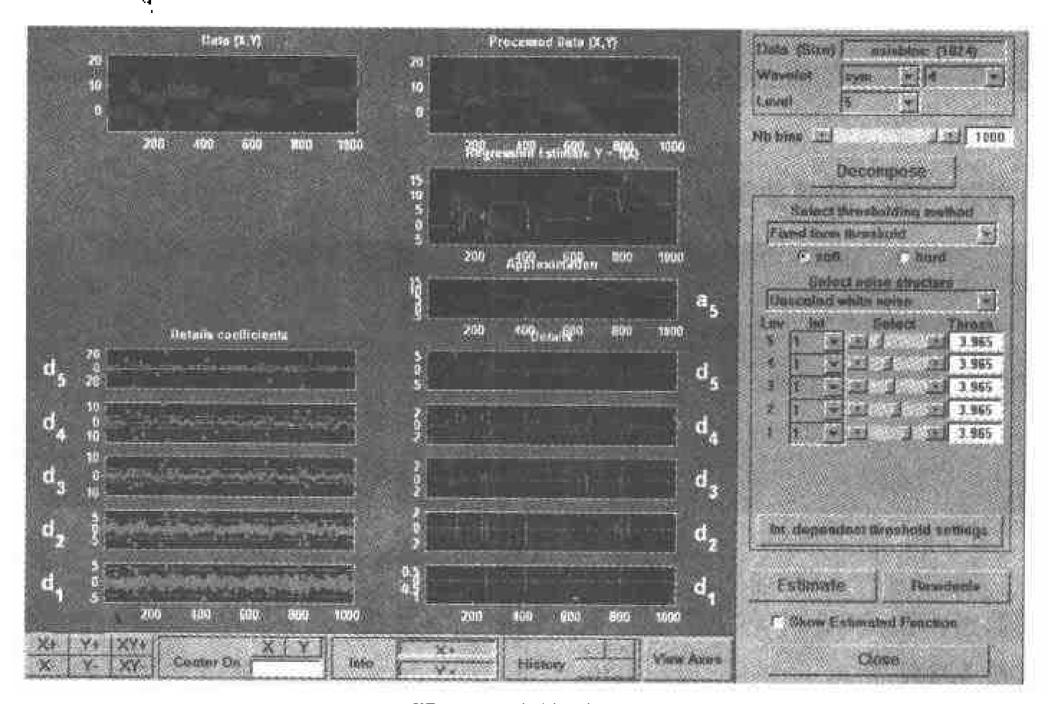


图 2-72 回归估计

可以看出，图 2-72 中黄线所示的回归估计信号是位于它下面的第 5 层的低频信号和经过阈值处理后的高频部分的和。用户可以通过图 2-72 右边的阈值选择框分别改变各层次的阈值。

下面我们针对随机间距观测上面回归估计，看看有什么不同。

2.7.2 一维随机间距观测估计（随机性设计 Stochastic Design）

通过以下方式进行一维随机间距观测估计：

（1）在图2-67中的【File】菜单里选择【Load Data for Stochastic Design Regression】命令，选择MATLAB安装目录下的toolbox/wavelet/wavedemo子目录下的ex1nsto.mat文件作为回归估计的信号。结果如图2-73所示。

“装仓”（binning）处理是必须的，因为它将对不规则间距信号X的回归估计问题转化为传统的确定性设计（fixed design scheme），使得快速小波变换能够应用。

（2）在图2-73中的右上角选择基本小波为sym4 和尺度数Level为5，Nb bins选择125。选择好以上参数后，就可以单击【Decompose】按钮，经过短时间的计算后将显示高频系数。

（3）在Select thresholding method下拉框中选择Penalize low，然后单击【Estimate】按钮，结果如图2-74所示。

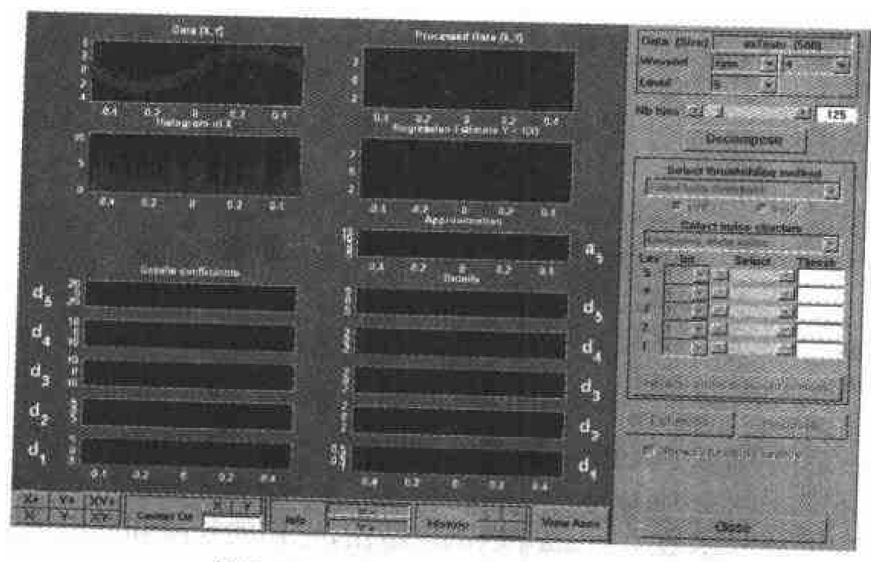


图 2-73　ex1nsto.mat 文件运行结果

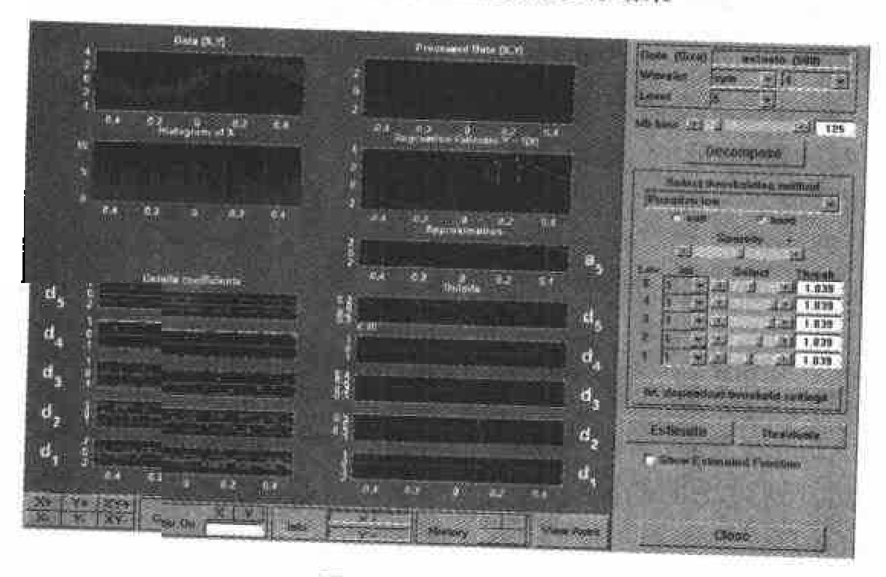


图 2-74　显示结果

（4）选择 Show Estirated Function，对比估计的效果如图 2-75 所示。

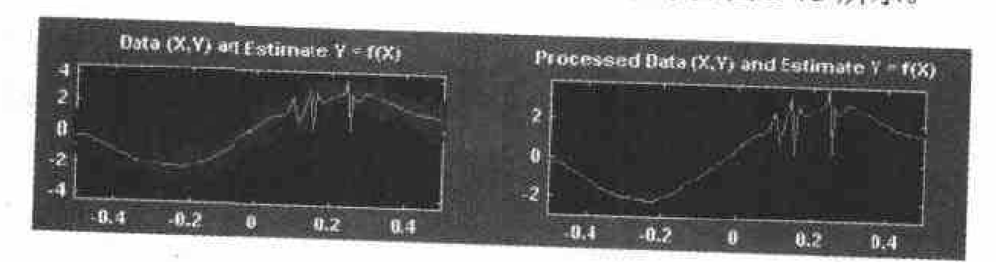


图 2-75　对比估计

2.7.3　回归估计专有图形工具中的信息交互

1．保存估计函数

在进行完回归估计后，用户可以以 MAT 格式的文件保存估计函数。在图 2-74 所示的

卡菜单【File】里选择【File】→【Save Function】命令，然后以 fex1nsto.mat 保存，并将其装载进工作空间：

```
>>load fex1nsto;
>>whos
  Name          Size       Bytes     Class
   X            96×96      73728     double array
  map           255×3      6120      double array
  thrParams     1×5        580       cell array
  valTHR        3×4        96        double array
  wname         1×4        8         char array
  xdata         1×125      1000      double array
  ydata         1×125      1000      double array
```

说明：可以看出，估计函数由 xdata 和 ydata 给出，它们的长度和我们上面选择的 Nb bins 数相等。与层次相关的阈值保存在变量 thrParams 里。

2、装载数据

这里要注意，装载进的文件至少要包含一个向量。当只包含一个向量时，系统将其作为 ydata，xdata 向量自动产生；当包含两个向量时，它们必须被称为 xdata 和 ydata 或 x 和 y，并且它们的尺度必须相等。例如，装载上面例子中的数据：

```
>>clear
>>load ex1nsto
>>whos
  Name       Size       Bytes     Class
  x          1×500      4000      double array
  y          1×500      4000      double array
```

在这一节最后，通过下面的命令将延拓模式改回到“零填补”模式：

```
dwtmode('zpd');
```

2.8 一维小波密度估计

密度估计是可靠性研究的关键之一。它可以给出某个厂家生产的电视机的全寿命概率分布，进行瞬时可用性计算，以及计算平均故障时间等很有用的参数。下面我们来看小波在这一领域中的应用。

2.8.1 密度估计概述

假设 $X(i)$（1δ *i*δ *n*）是由某一密度未知的分布采样而来，我们来寻求对其密度进行估计。

我们知道，直方图可以表征一系列测量值的密度分布信息。在 19 世纪初，法国科学家拉普拉斯通过重复进行相同数量的观测，给出一个简单的函数来计算这些测量值的密度分布。这个函数现在被称为拉普拉斯－高斯分布。

分析证明，当密度函数 $h(x)$ 具有不规则性时（例如有一个断点或其一阶导数有断点），

采用小波进行密度函数估计是一个好的解决方法。

基本的思想是将其转化为一个确定性回归模型，步骤如下：

（1）首先将采用信号 *X* 转化为（*Xb*，*Yb*），这里 *Xb* 进行直方图划分过程等间距分布，*Yb*（*i*）是 *X* 落在第 *i* 个矩形块里的采样点数。

（2）采用快速算法将 *Yb* 作为一个信号进行小波分解，这里默认的 *Xb* 是 1, 2, …, *nb*（*nb* 是柱条数）。

（3）对分解后的系数进行阈值处理。

（4）从上述处理后的系数重构密度函数 *h* 的一个估计 *h*1。

（5）对 *h*1 进行后续处理。

步骤（2）和（4）是标准的小波变换与重构。步骤（1）依赖于柱条数 *nb*（the number of bins），*nb* 可看做带宽参数。在密度估计中，*nb* 一般比原始观测数目（*X* 的信号长度）小，典型的默认值是 length(*X*) / 4。

有关利用小波进行密度估计的更详细的讨论请参考有关文献，下面我们用 MATLAB 6.5 小波工具箱中的密度估计图形工具（Density Estimation Graphical User Interface）来完成上述几个步骤，使读者对其有一个感性的认识。

在进行下面的分析之前，首先通过下面的命令将延拓模式切换到“对称填充”（symmetric padding）：

```
>>dwtmode('sym');
```

2.8.2　一维小波密度估计图形工具的使用

1．启动一维小波密度估计图形工具

在图 2-4 所示的小波工具箱主菜单中选择 Density Estimation 1-D，出现图 2-76 所示的一维小波密度估计图形工具。

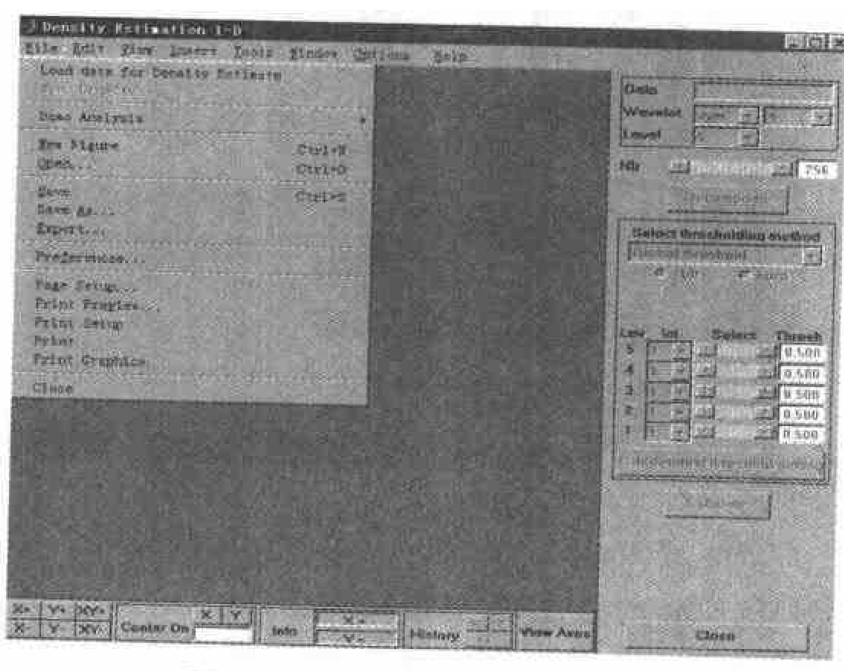


图 2-76　一维小波密度估计图形工具

2．装载数据

在图 2-76 中单击菜单命令【File】→【Load Data for Density Estimate】，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 ex1cusp1.mat 文件。如图 2-77 所示。

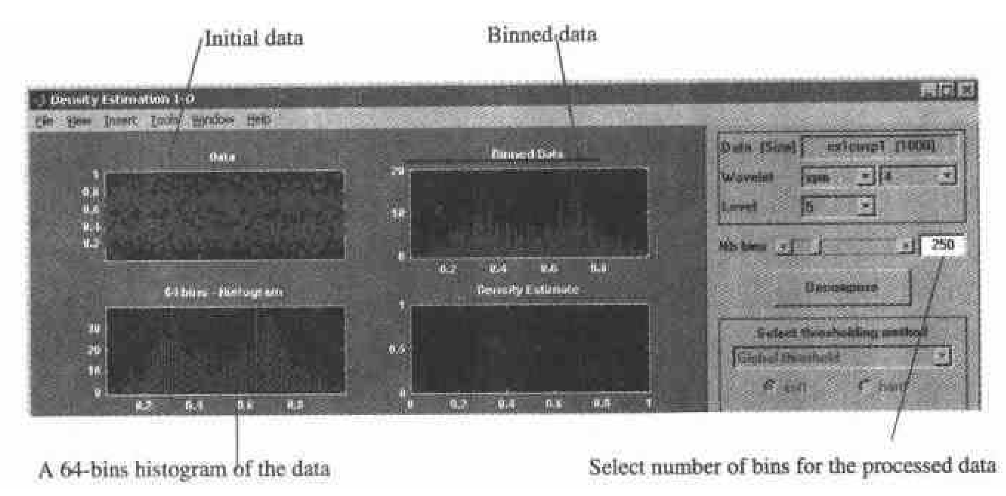


图 2-77　ex1cusp1.mat 文件运行结果

可以看出，这里我们取默认的柱条数为 250，经过直方图划分处理后的数据将在下面进行小波分解。

3．对经过直方图划分处理后的信号（binned data）进行小波分解

在图 2-76 中的右上角选择基本小波为 sym6 和尺度数 Level 为 4。选择好以上参数后，就可以单击【Decompose】按钮，经过短时间的计算后将出现如图 2-78 所示的分解结果。

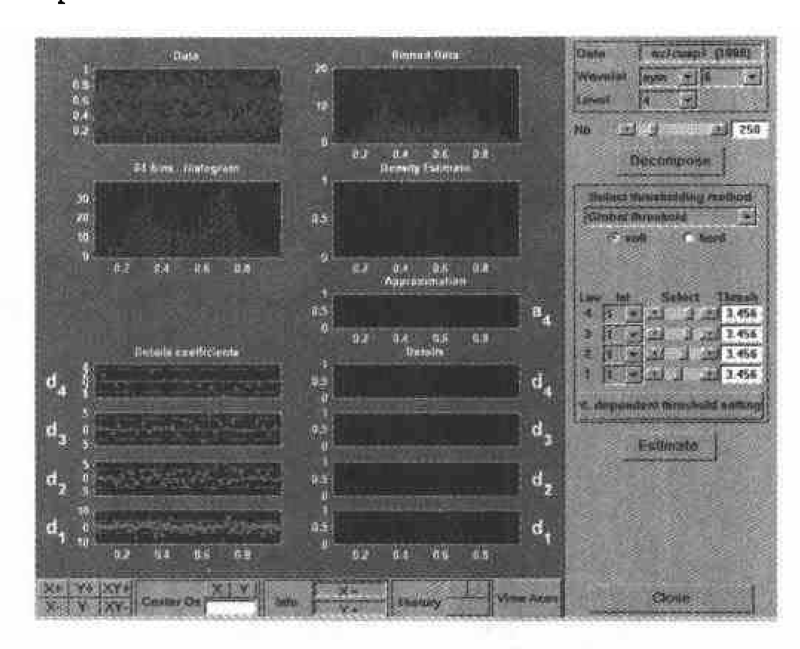


图 2-78　小波分解

4．进行密度估计

这里我们就取图 2-79 所示的系统默认的全局软阈值。用户也可根据实际情况通过窗口右边的滑动条来指示和调整相应各层次的阈值，结果以黄点线反映在窗口左边的图形

中。选择好阈值后，单击【Estimate】按钮，结果如图 2-79 所示。

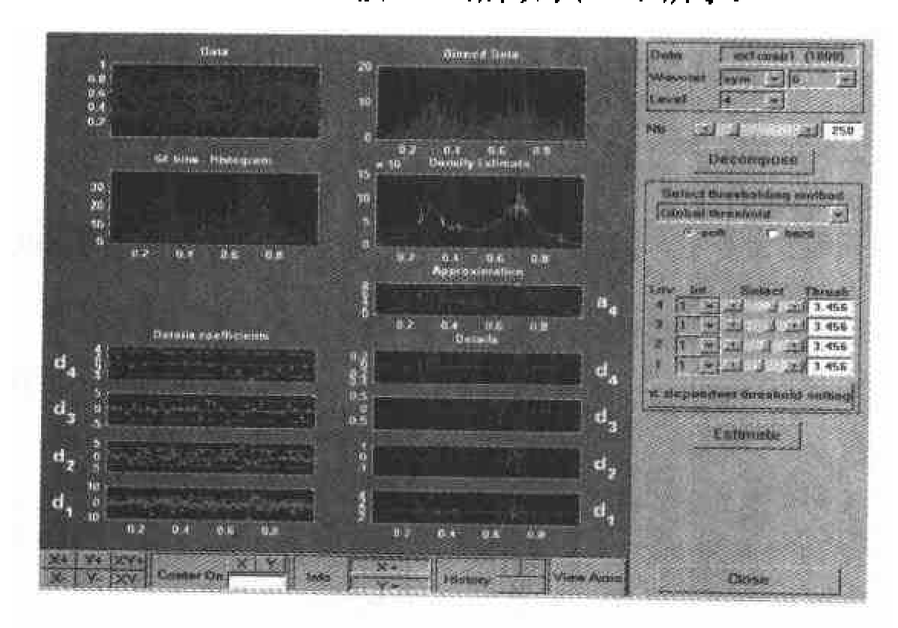


图 2-79 密度估计 1

可以看出，图 2-79 中黄色的密度曲线很不规则，它是由图 2-80 所示的低频信号 a4 和通过阈值处理后的系数重构得到的高频信号的叠加。

下面我们再通过图 2-79 右边的 Select thresholding method 下拉框重新选择 By level threshold2。再单击【Estimate】按钮，估计结果如图 2-80 所示。

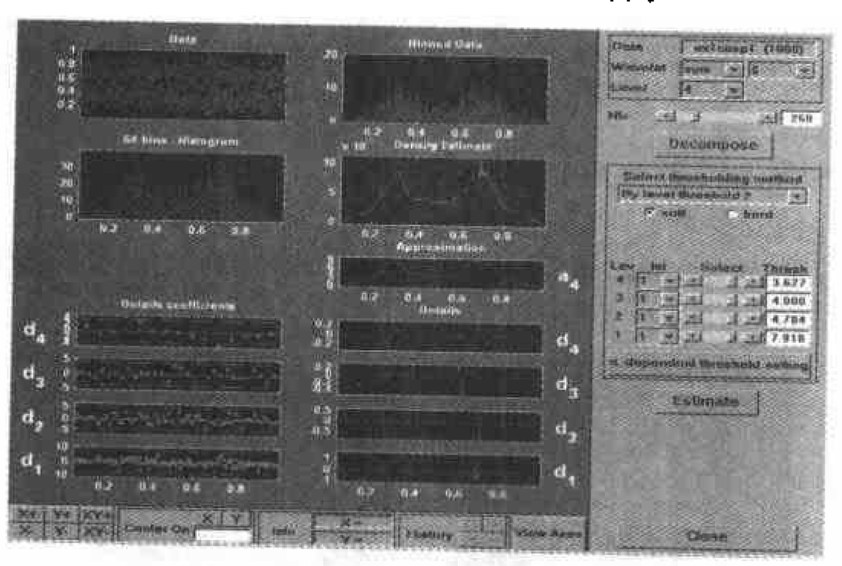


图 2-80 密度估计 2

可以看出，估计结果比上面的要好，它正确识别出了密度的平滑部分和在 0.7 处的尖峰。

5. 一维小波密度估计图形工具中的信息交互

用户可以通过选择图 2-80 所示的主菜单中【File】→【Save Density】的命令，来保存估计得到的密度分布，在弹出的对话框里输入 dex1cusp，再将其装载进工作空间：

```
>>load dex1cusp
>>whos
  Name          Size          Bytes          Class
```

thrParams	1×4	464	cell array
wname	1×4	8	char array
xdata	1×250	2000	double array
ydata	1×250	2000	double array

密度分布由变量 xdata 和 ydata 给出，它们的长度和上面估计过程中选取的柱条数一样。变量 thrParams 的含义与前几节中的含义一样。

最后，通过下面的命令将延拓模式（extension mode）该为 zero padding：

```
>>dwtmode('zpd')
```

2.9 一维小波系数的自适应阈值处理

这一节将让读者掌握针对一维信号小波系数的局部阈值处理。这种自适应阈值处理可以让用户分层定义与时间相关（time-dependent）的阈值限，以改进针对含有非平稳噪声时的消噪效果。这里我们以上述的一维离散平稳小波消噪图形工具为例来说明怎样进行自适应阈值处理，其他图形工具中的用法类似。

2.9.1 一维局部阈值处理图形工具——消噪

1. 启动 SWT De-noising 1-D 并装载数据

在图 2-4 所示的小波分析图形工具箱主界面中选择 SWT De-noising 1-D，单击菜单命令【File】→【Load Signal】，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 nblocr1.mat 文件。

选择基本小波为 db1，层次数 Level 为 5，然后单击【Decompose Signal】按钮，过一会将显示出平稳小波变换的低频和高频显示的图形表示。

采用系统默认的阈值设置：固定软阈值 Fixed form soft 和尺度未知的白噪声 Unscaled white noise，单击【De-noise】消噪按钮。效果如图 2-81 所示。

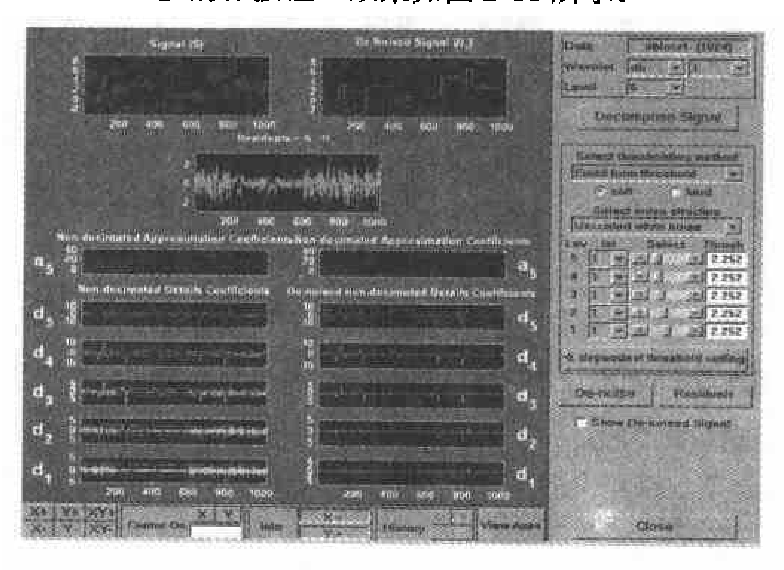


图 2-81 阈值设置

总体看消噪效果较好，但当含噪信号很不规则时显得过于平滑。

再在图 2-81 中选择硬阈值（hard）代替上面采用的软阈值，重新单击【De-noise】消噪，结果如图 2-82 所示。

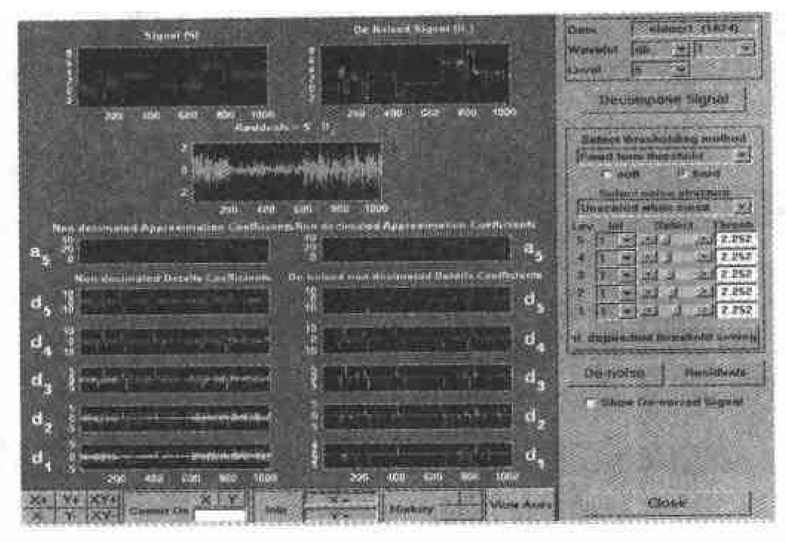


图 2-82　消噪

可以看出，消噪效果并不理想，信号仍然含有明显的噪声。这说明当噪声是时变的时候，传统的消噪方法效果很有限。

2．产生与时间间隔相关的阈值

单击图 2-82 中的【Int. dependent threshold settings】按钮，出现图 2-83 所示的阈值设置窗口。

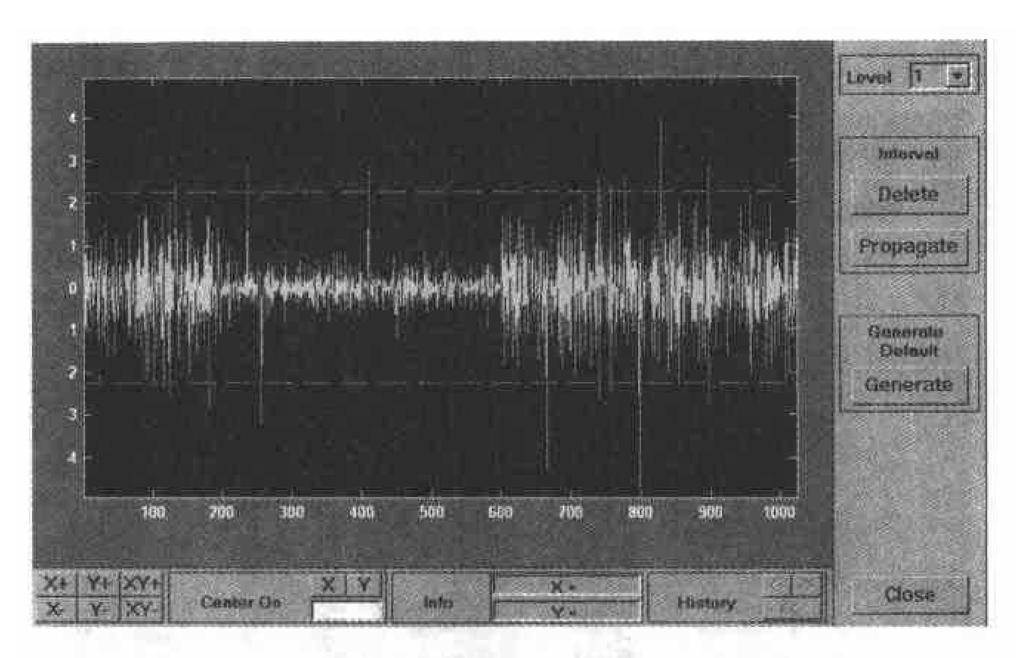


图 2-83　阈值设置窗口

单击【Generate】按钮，则经过短暂的计算后，在图 2-84 中显示出与时间间隔相关的比较合适的阈值。

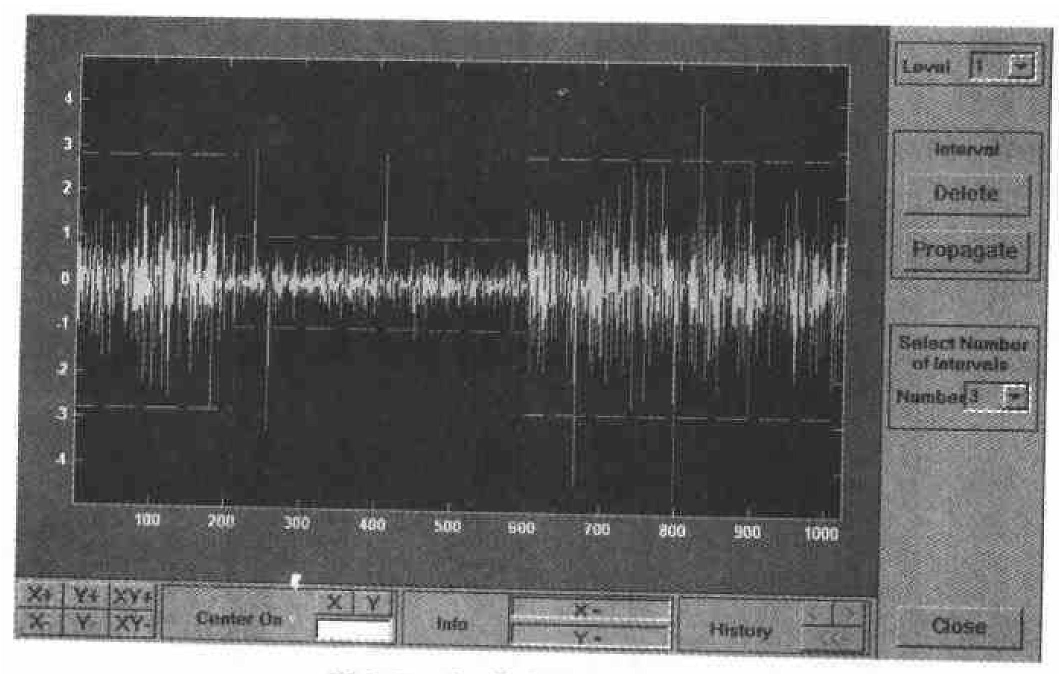


图 2-84 与时间间隔相关的阈值

可以看出，在时间轴上两条黄线将阈值分为三段进行设置，这是由于各段的变换情况差异很大，这样可以取得最优的消噪效果。不过用户也可在图 2-84 右边的 Select Number of Intervals 下拉框中重新选择分段数（从 1 到 6）。这样系统将根据新的分段数重新分段设置阈值。这里我们采用系统默认的值 3。

单击图 2-84 中的【Close】按钮关闭上述窗口，会出现阈值更新（Update thresholds）对话框，单击【Yes】按钮。则随后一维平稳小波消噪窗口也将更新。用户可以在图 2-82 中左边的 d1 到 d5 的图形表示中看到有两条红线分隔符将时间轴划分为三段。然后单击消噪按钮，结果如图 2-85 所示（彩色效果图见彩插 3）。

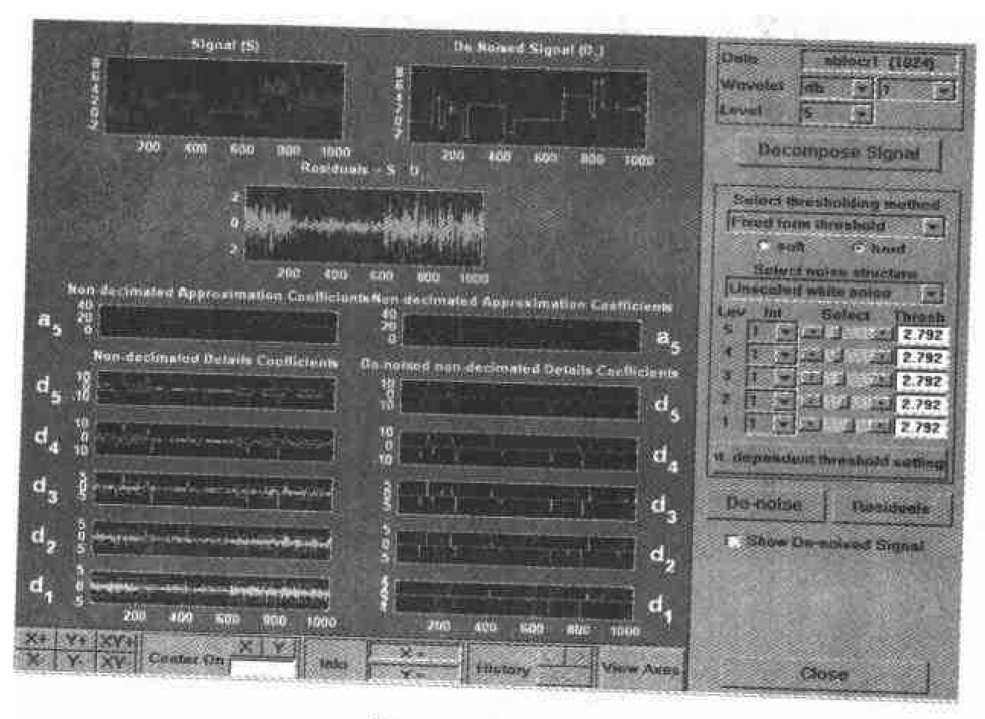


图 2-85 阈值更新

可以看出，消噪效果比较理想，但仍然含有一些毛刺。

3．修改与时间间隔相关的阈值

可以增加阈值以只保留每一层次值最大的小波系数。可以通过拖动 2-85 图左边的图形中黄色的阈值线或直接在右边各层次阈值设置框中选择来达到上面的目的。注意，也可直接在左边图中按住鼠标左键拖动红色的分界线来改变时间间隔。

用户也可在图 2-83 中自己定义与时间间隔相关的阈值。首先单击【Delete】按钮，红色的分界符消失，然后双击鼠标左键来设置新的分界符。通过选择 Level 下拉框中的层次值对每一层次的阈值分别调剂，也可通过单击按钮【Propagate】将当前层次的分界符设置应用到所有的层次上去。有三点值得注意：

- 再一次用鼠标左键双击某个分界符将会删除它；
- 可以通过在垂直方向上的红色时间间隔分界符和水平方向上的表示阈值的黄色点线上，按住鼠标左键不放来拖动它们；
- 每一层最大的时间间隔数是 10。

另外用户可以通过图 2-55 所示的一维平稳小波消噪窗口中的菜单【File】→【Demo Analysis - Interval Dependent Thresholds】来学习采用时间间隔相关的阈值进行消噪。例如用户可以选择最后一项【with haar at level 4】→【Elec. consumption –3 intervals】命令，系统将会自动给用户演示整个分析过程。

2.9.2　采用按时间间隔设置阈值进行消噪后的信息保存与装载

在进行完上面的步骤后，用户可以通过菜单命令【File】→【Save De-noised Signal】保存消噪处理后的信号。假设我们以文件名 dnelec.mat 保存上面系统自带的一个演示例子中经过消噪处理后的电力消耗信号，将其装载到工作空间：

```
>>load dnelec
>>whos
  Name         Size       Bytes     Class
  dnelec       1×2000     16000     double array
  thrParams    1×4        656       cell array
  wname        1×4        8         char array
  xdata        1×250      2000      double array
  ydata        1×250      2000      double array
```

这里与前面不同的在于变量 thrParams。ThrParams{i}是一个 nbint×3 的数组（nbint 是上面划分的时间间隔数，这里为 3），每一行包含了阈值的下限、上限和所取阈值。例如对第一层（level 1）有：

```
>>thrParams{1}
ans =
1.0e+03 *
0.0010 0.0980 0.0060
0.0980 1.1240 0.0204
1.1240 2.0000 0.0049
```

2.10 小波系数选取

本节讲述利用 MATLAB 6.5 小波分析工具箱中的专用工具来进行小波系数的选择。这些工具提供了一个图形接口工具来探究基于多种小波系数选取方案的重构策略。

2.10.1 一维离散小波系数选择——图形接口方式

和前两节一样，首先通过下面的命令将延拓模式切换到“对称填充”方式：

```
>>dwtmode('sym');
```

1．启动一维离散小波系数选择图形工具

在图 2-4 所示的小波分析图形工具主菜单里选择 Wavelet Coefficient Selection 1-D，则出现图 2-86 所示的一维离散小波系数选择窗口。

2．装载信号

在图 2-86 中单击菜单命令【File】→【Load Signal】，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录下的 noisbump.mat 文件。

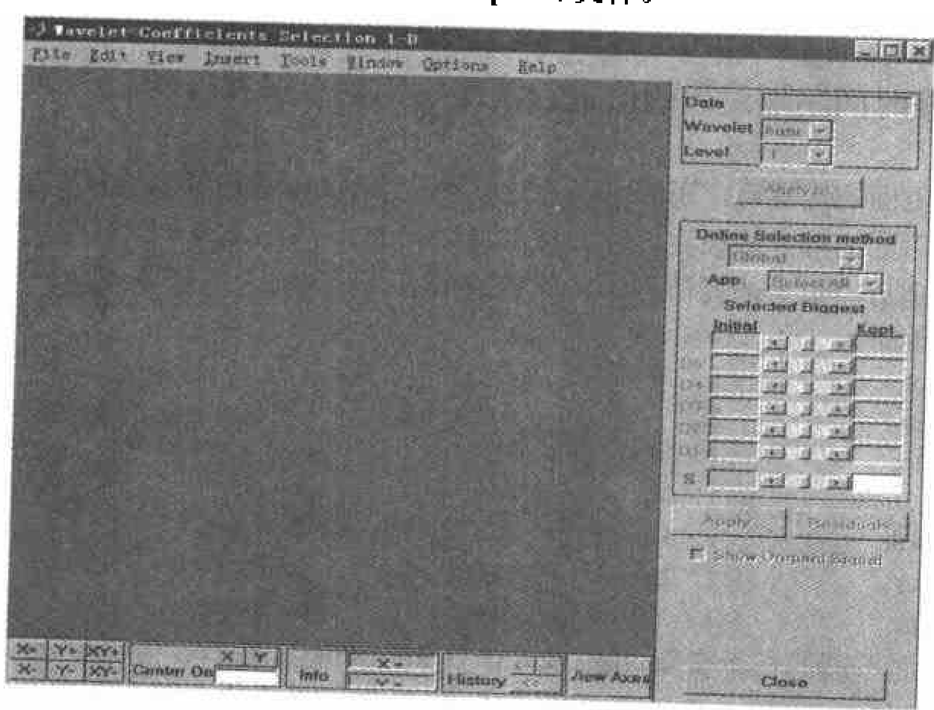


图 2-86 一维离散小波系数选择窗口

3．执行小波分解

在图 2-86 中的右上角选择基本小波为 db3 和尺度数 Level 为 6。选择好以上参数后，就可以单击【Analyze】按钮。经过短时间的计算后将出现如图 2-87 所示的分解结果。

从图 2-87 可以看出，在原始信号下面是它的小波分解低频系数 A6 和高频系数 D6、D5、D4、D3、D2、D1。在合成信号（synthesized signal）下面是选取的小波系数，在这

一步和左边的初始小波系数相同，因为还没有对初始小波系数进行选取，保留了所有的初始小波系数。

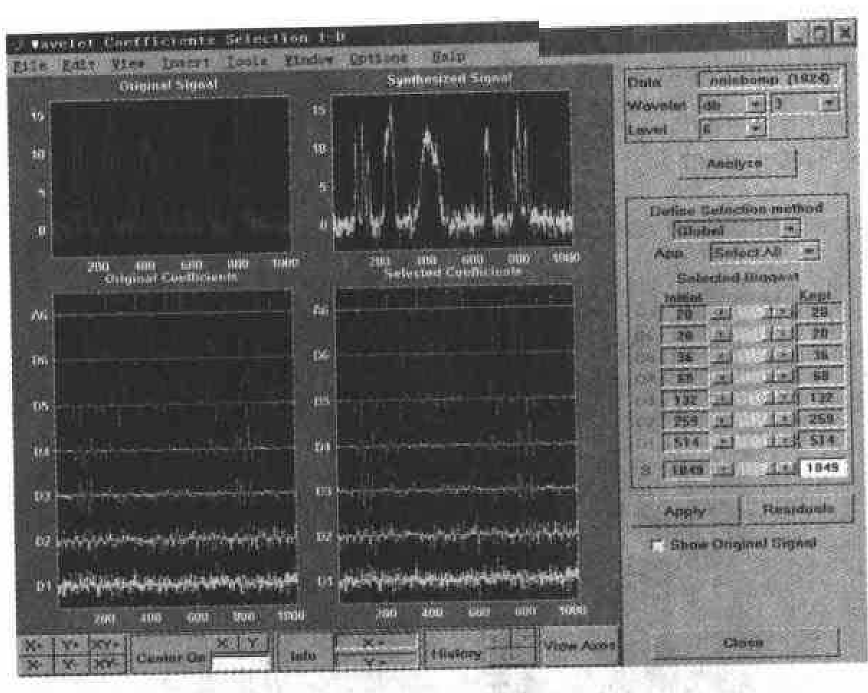


图 2-87　小波分解

4．对小波系数进行全局最大值选择

在图 2-87 窗口右边有一列 Kept 值，最底端显示所有的系数个数是 1049，这比原始数据 1024 稍多一点。用户可以选择在 1049 所在的编辑框中输入新的最大值系数选取个数。这里我们输入 40 并回车使其生效，可以看出上面每层选取的最大值系数个数也相应地自动更新（这里不能改手动，因为选取方法是全局选取）。然后单击【Apply】按钮，结果选取的系数如图 2-88 所示。

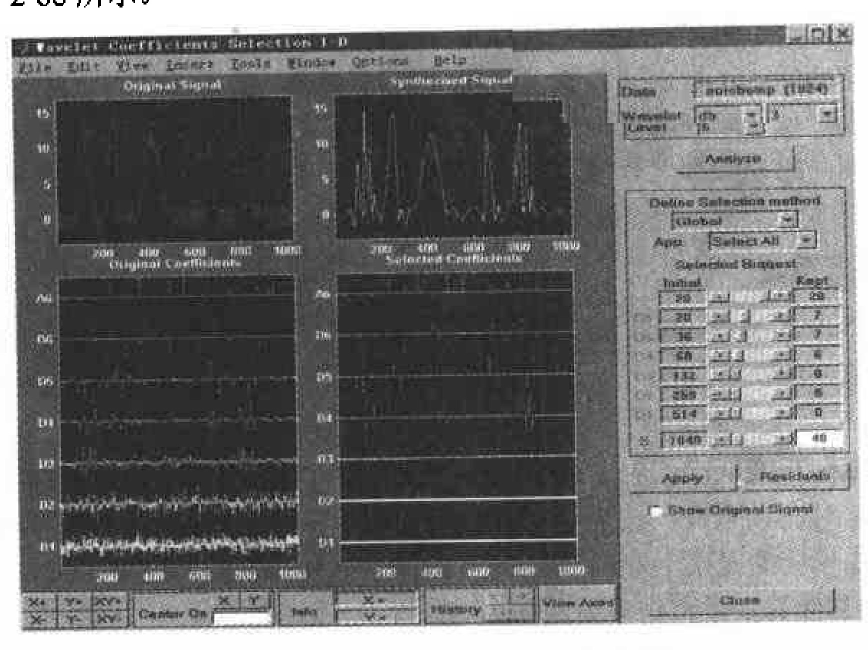


图 2-88　对小波系数进行全局最大值选择

从图 2-88 可以看出，选取后的小波系数只保留了 40 个，其中低频部分 A6 保留了 20，高频部分 D6、D5 分别保留了 7 个，D4 保留了 6 个。

在上面的步骤中，可以看出初始的 20 个低频小波系数被全部保留下来。这是由于在窗口右边的 App.（Approximation Coefficients）下拉框里选取了 Select All，用户可以改变该项选择。我们选择 Unselect（即全部舍弃），再单击【Apply】按钮，可以从图 2-89 看出所有的低频小波系数全部被舍弃。

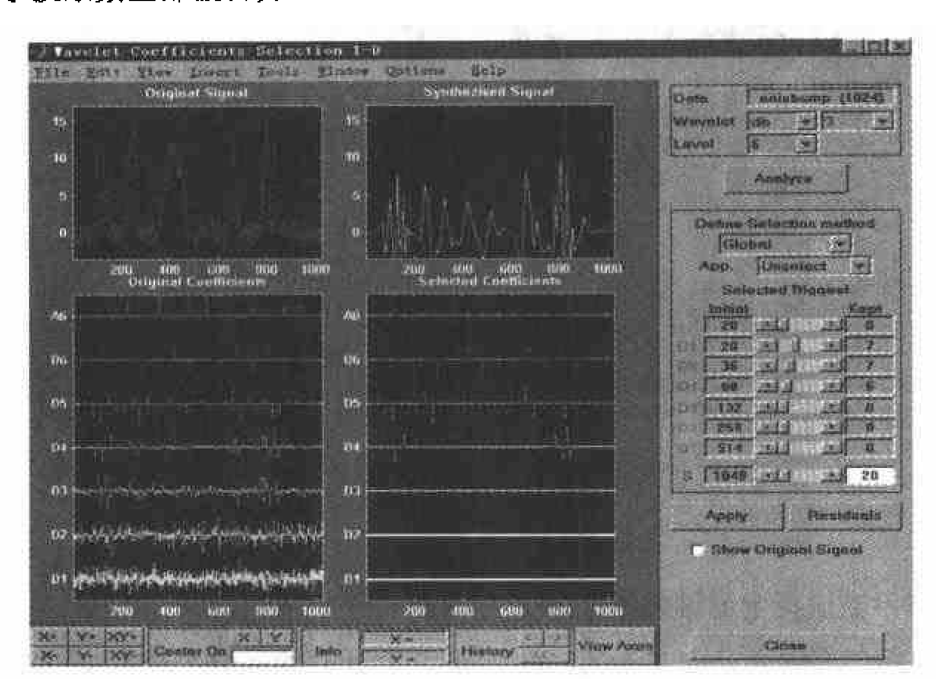


图 2-89　舍弃低频小波系数

用户还可以选择 Selectable，在最大值系数选取个数编辑框里输入 80 并回车，然后单击【Apply】按钮，就可以保留部分低频小波系数（这里是 15）。

5．分层选择最大值系数

在 Define Selection method 下拉框里选择 By Level，就可以针对分解的每一层次分别定义最大值系数选取个数（通过拖动滑动条或直接在 Kept 编辑框里输入数值）。这里我们对各层次低频和高频系数都选择 4，然后单击【Apply】按钮使其生效，结果如图 2-90 所示。

6．手动选择系数

在 Define Selection method 下拉框里选择 Manual，则在窗口左边显示出原始信号及其分解后小波系数，但在右边刚开始没有系数被保留，因而合成信号 synthesized signal 也是空的。

分别通过鼠标左键双击代表系数的线条来选取 7 个系数。对低频系数，被选取的系数线条由蓝变黄；对高频系数，被选取的系数线条由绿变黄。如果用户想放弃某个选取的系数，只需重新在相应的线条上双击即可。选取的 7 个系数以黄色线条出现在窗口中部，然后单击【Apply】使其生效，结果如图 2-91 所示。

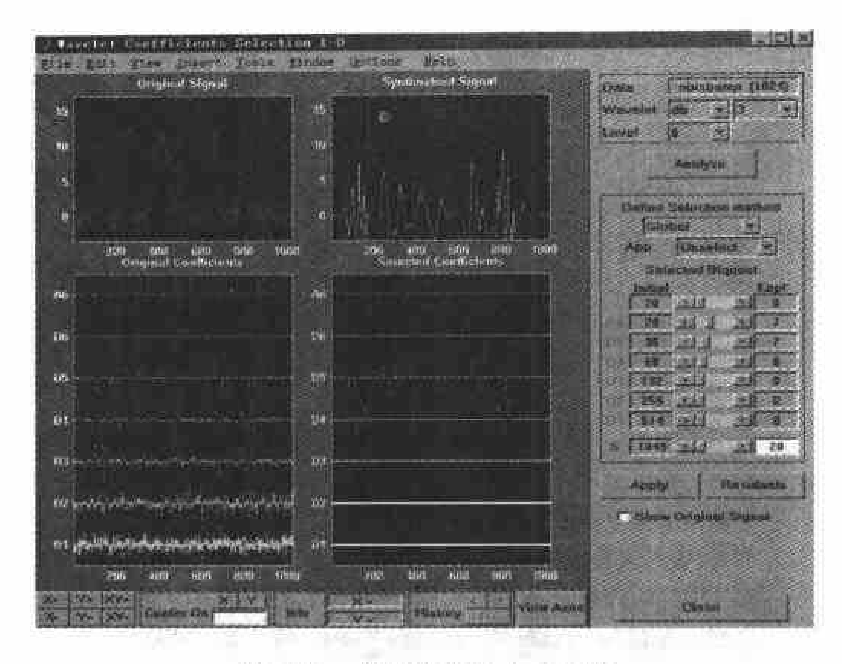


图 2-90　分层选择最大值系数

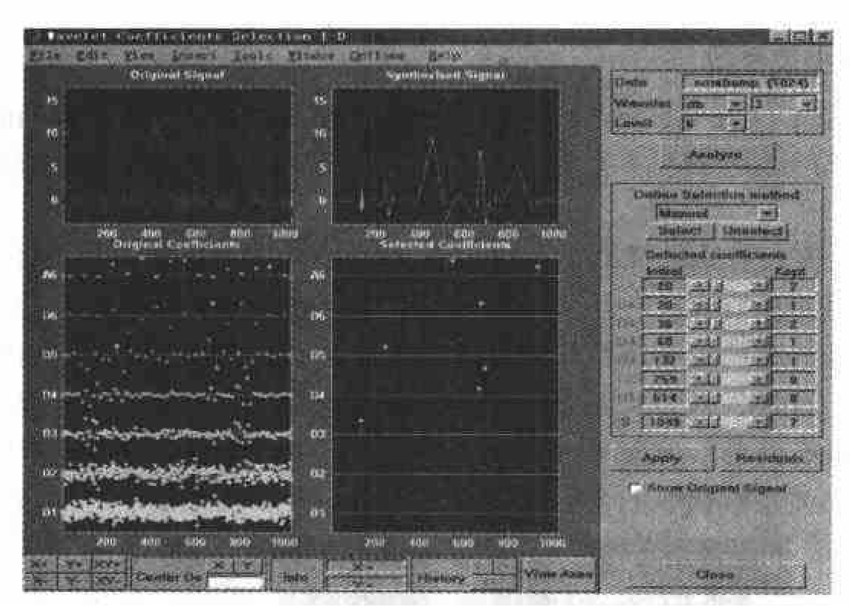


图 2-91　手动选择系数 1

用户也可以通过选择框来选取或取消已经选取的一组系数：按住鼠标左键不放，在初始系数或选取的系数图形上拖动一个矩形框包含已经选取的系数，然后单击窗口右边的【Unselect】按钮，再单击【Apply】按钮，则所有已经选取的系数被取消，合成信号又变成空的信号。当代表各个小波系数的线条很近时，用户可以在选取或取消选取前放大系数的图形表示，以准确地进行操作。

在系数图形表示的横坐标 800 位置附近拖动一个矩形选择框，然后单击右边的【Select】按钮，可以看到矩形选择框内的系数变成了黄色，表示被选中。然后单击【Apply】按钮，结果如图 2-92 所示。

从图 2-92 可以看出，信号在位置 800 附近被完美地重构出来，用户可以选择图 2-92 右下部的 Show Original Signal 复选框，使位置 800 附近的初始信号和合成后的信号同时

显示在一个图中进行对比验证。这也说明，小波分析具有局部分析功能。

图 2-92　手动选择系数 2

7．自动选取小波系数

在图 2-92 中的 Defien Selection method 下拉框里重新选择 Stepwise movie（步进选择演示），则显示出与手动选择的初始窗口除左边有所不同外其余几乎一样的窗口。如图 2-93 所示。

图 2-93 所示窗口右边选择框里，Min 编辑框里输入的数字代表小波系数选取的最小个数，Step 编辑框里的数字代表每次选取小波系数个数递增的步长，Max 编辑框里输入的数字代表小波系数选取的最大个数。这里我们将 Min 取 1，Max 取 31，Step 取 1，然后使下面的 AutoPlay 有效，这样系统会自动演示选取小波系数的整个动态过程。这时单击【Start】按钮，系统就会从选取第一个系数开始直到选够 31 个系数为止，在这个过程中重构信号也随着小波系数选取的个数增加而越来越接近于初始信号。

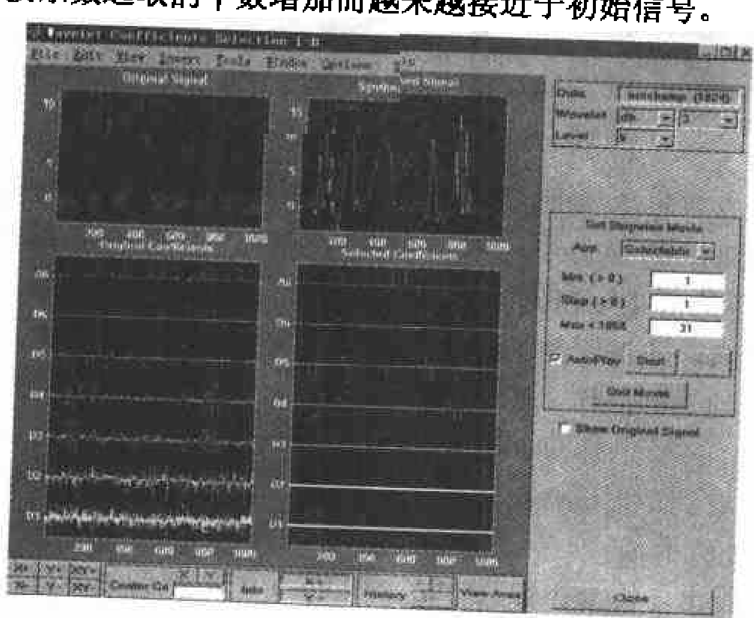


图 2-93　自动选取小波系数



8．保存合成后的信号

从图 2-93 所示的主菜单里选择【File】→【Save Synthesized Signal】菜单命令，可以以 MAT 格式的文件保存合成后的信号以进行后续的处理。

在这节最后，将延拓模式切换到“零填充”方式：

```
>>dwtmode('zpd');
```

2.10.2　二维离散小波系数选择——图形接口方式

这一小节讲述利用 MATLAB 6.5 小波分析工具箱中的专用工具来进行二维小波系数的选择。该工具以图形方式来探究基于多种小波系数选取方案的重构策略。

由于该图形工具基本和上一小节的一维小波系数选择类似，所以很多相同的地方不再解释。

首先通过下面的命令将延拓模式切换到“对称填充”方式：

```
>>dwtmode('sym');
```

1．启动一维离散小波系数选择图形工具

在图 2-4 所示的小波分析图形工具主菜单里单击 Wavelet Coefficient Selection 2-D，则出现图 2-94 所示的二维离散图像小波系数选择图形窗口。

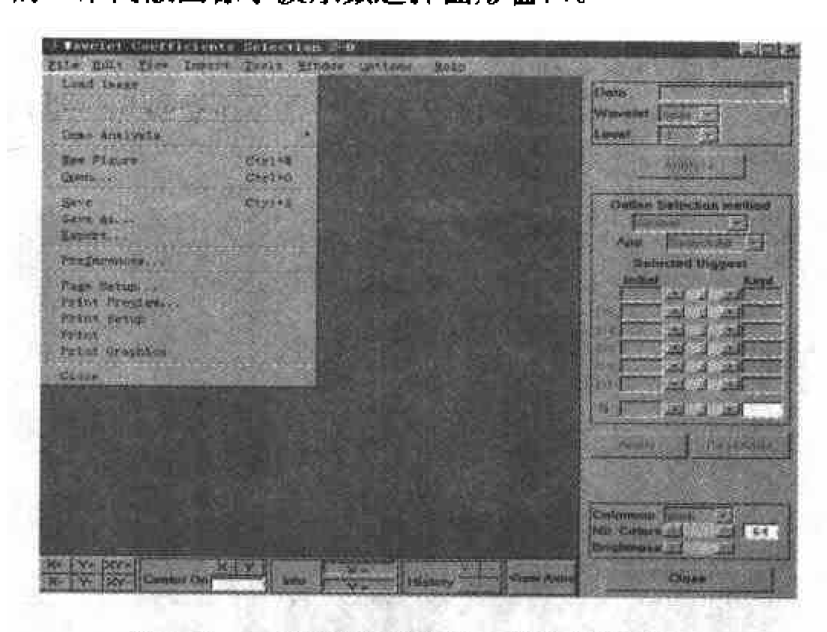


图 2-94　二维离散图像小波系数选择图形窗口

2．装载信号

在图 2-94 中单击菜单命令【File】→【Load Image】，选择 MATLAB 6.5 安装目录下 toolbox/wavelet/wavedemo 子目录中的 noiswom.mat 文件。

3．执行图像的小波分解

在图 2-94 中的右上角选择基本小波为 sym4 和尺度数 Level 为 4。选择好以上参数后，就可以单击【Analyze】按钮。经过短时间的计算后将出现如图 2-95 所示的分解结果。

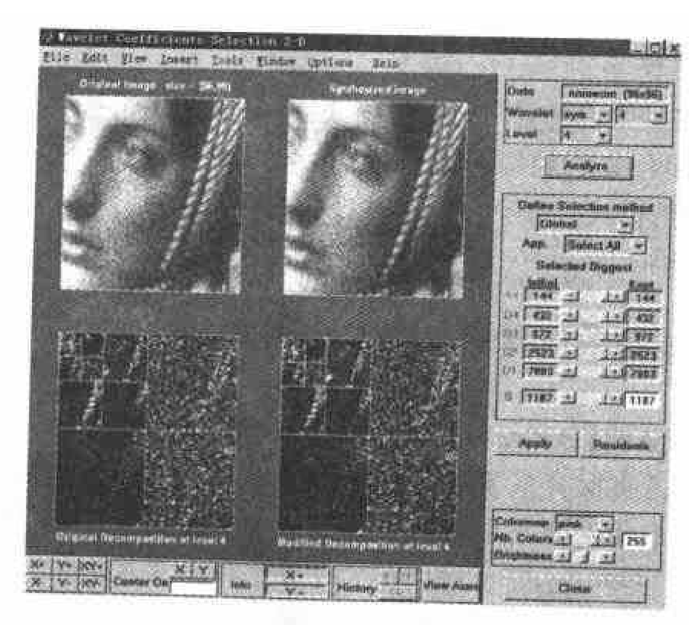


图 2-95　图像的小波分解 1

从图 2-95 可以看出，在原始图像下面是它的小波分解，在合成图像（synthesized image）下面是选取的小波系数，此时和左边的初始小波系数相同，因为还没有对初始小波系数进行筛选，保留了所有的初始小波系数。这里共有 11874 个系数，比初始图像的像素点（96×96=9216）要多一些。那些在 9216 和 11874 之间的系数来自采用当前的延拓模式而进行的额外的离散小波变换 DWT。详细情况可参见下一节的信号延拓部分。

在图 2-95 窗口右边的 Kept 标签下的最大值系数选取个数编辑框里输入 1100 并回车，然后单击【Apply】按钮，结果如图 2-96 所示。

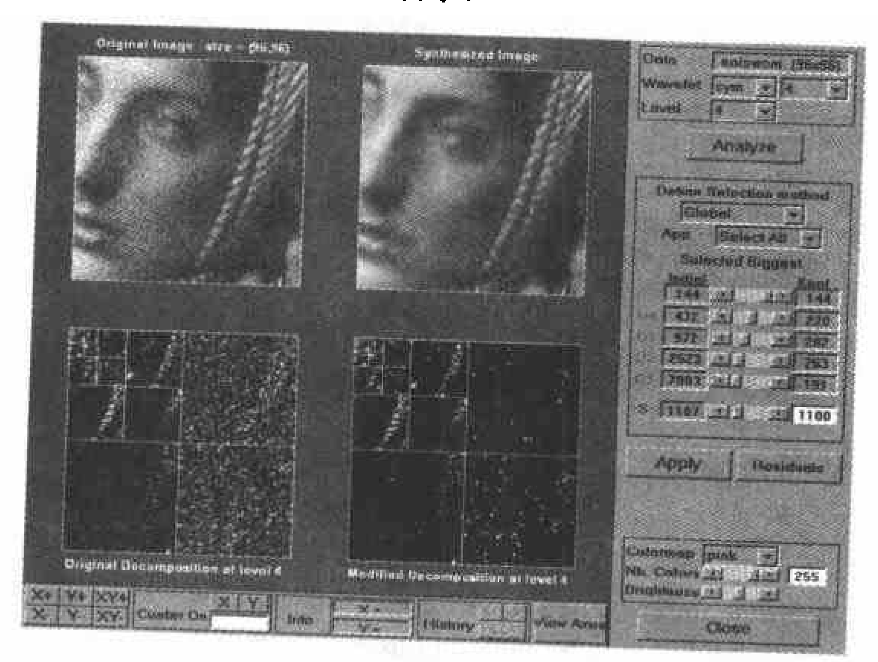


图 2-96　图像的小波分解 2

在上面的步骤中，可以看出初始的 144 个低频小波系数被全部保留下来。这是由于在窗口右边的 App.（Approximation Coefficients）下拉框里选取了 Select All，用户可以改变该项选择。

4. **分层选择最大值系数**

在 Define Selection method 下拉框里选择 By Level，就可以针对分解的每一层次分别定义最大值系数选取个数（通过拖动滑动条或直接在 Kept 编辑框里输入数值）。这里我们对各高频系数都选择 100，然后单击【Apply】按钮使其生效，结果如图 2-97 所示。

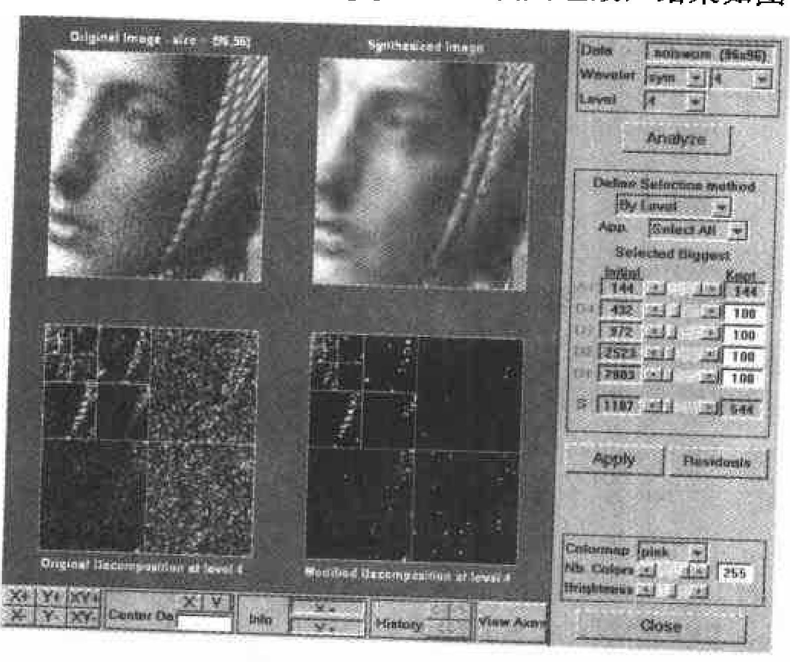


图 2-97　分层选择最大值系数

5. **自动选取小波系数**

在图 2-97 中的 Defien Selection method 菜单里重新选择 Stepwise movie（逐步选择演示），这里我们将 Min 取 144，Max 取 1500，Step 取 20，然后单击【Start】按钮。系统会自动演示选取小波系数的整个动态过程，在窗口下面也动态地显示目前已经选取的小波系数个数。如图 2-98 所示。

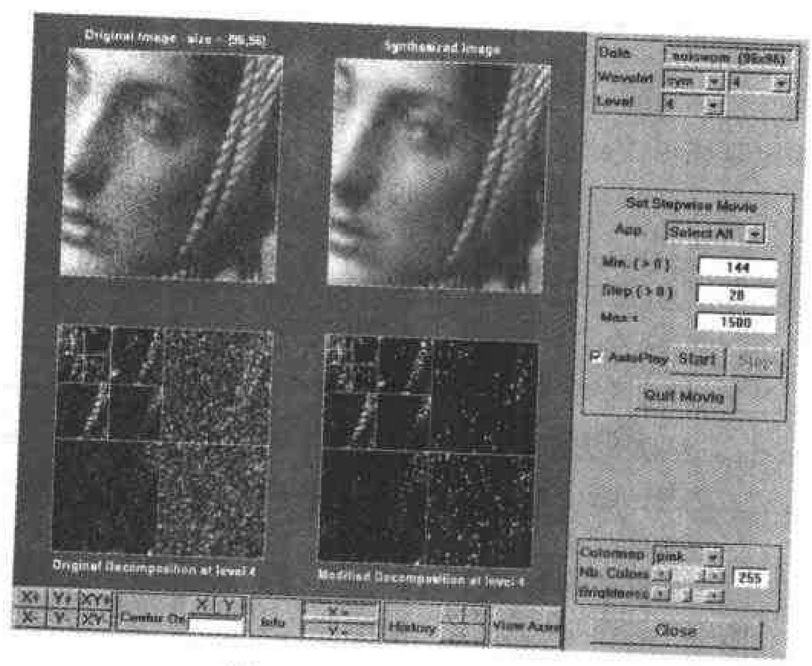


图 2-98　自动选取小波系数



在动态显示过程中，当用户对合成的图像比较满意时，也可以随时单击【Stop】按钮来人为停止选取。

6．保存合成后的信号

和前一小节一样，这里不再赘述。

最后记住也要将延拓模式切换到“零填充”方式：

```
>>dwtmode('zpd');
```

2.11　一维信号延拓或截断

在实际中，由于种种原因，有时需要对信号进行延拓或截断，因此需要掌握如何利用小波工具箱进行一维延拓或截断。

这一节讲述如何利用小波工具箱进行一维延拓或截断。

2.11.1　一维延拓——命令行方式

用于信号延拓的小波工具箱函数只有一个：wextend。其格式为：

① Y = wextend(TYPE,MODE,X,L,LOC)

② Y = wextend(TYPE,MODE,X,L)

参数 TYPE 决定延拓类型，如表 2-3 所示。

表 2-3　延拓类型

TYPE 值	含　义
1，'1'，'1d'或'1D'	一维延拓
2，'2'，'2d'或'2D'	二维延拓
'ar'或'addrow'	添加行
'ac'或 addcol'	添加列

参数 MODE 决定延拓模式，如表 2-4 所示。

表 2-4　延拓模式

MODE 值	含　义
'zpd'	零延拓 Zero extension
'sp0'	0 阶平滑延拓 Smooth extension of order 0
'sp1'	1 阶平滑延拓 Smooth extension of order 0
'sym'	对称延拓 Symmetric extension
'ppd'	周期性延拓 1
'per'	周期性延拓 2（当信号长度是奇数时，wextend 在右边添加一个和最后一个采样值一样的值，然后再按照 ppd 的模式进行延拓）

（1）当 TYPE = {1, '1', '1d'　或'1D'}时：

LOC='l'（或'u'）是进行左（或上）延拓；

LOC='r'（或'd'）是进行右（或下）延拓；

LOC='b'是进行双边延拓；

LOC='n'是进行空延拓；

LOC 默认值是'b'，此时参数 L 是延拓长度。

（2）当 TYPE = {'ar', 'addrow'}时，LOC 是一维延拓位置，默认是'b'，这时 L 是添加的行数。

（3）当 TYPE = {'ac', 'addcol'}时，LOC 是一维延拓位置，默认是'b'，这时 L 是添加的列数。

（4）当 TYPE = {2, '2', '2d' 或 '2D'}时，LOC = [LOCROW,LOCCOL]，这里 LOCROW 和 LOCCOL 都是一维延拓位置，默认的 LOC='bb'。L = [LROW,LCOL]，这里 LROW 是添加的行数，LCOL 是添加的列数。

参看例程 2-13 可以加深对此的理解。

例程 2-13

```
x = [1 2 3]  %原始信号
x =
1 2 3
l = 2;    % 一维延拓长度
xextzpd1 = wextend('1','zpd',x,l)  %一维零填充
xextzpd1 =
0 0 1 2 3 0 0
xextzpd2 = wextend('1D','zpd',x,l,'b')
xextzpd2 =
0 0 1 2 3 0 0
xextsym = wextend('1D','sym',x,l)  % 一维对称延拓
xextsym =
2 1 1 2 3 3 2
xextper = wextend('1D','per',x,l)   % 一维周期延拓
xextper =
3 3 1 2 3 3 1 2
X = [1 2 3;4 5 6]  % 原始图像
X =
1 2 3
4 5 6
l = 2;    % 二维延拓长度
Xextzpd = wextend(2,'zpd',X,l)  % 二维零延拓
Xextzpd =
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 1 2 3 0 0
0 0 4 5 6 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

```
Xextsym = wextend('2D','sym',X,1)     %二维对称延拓
Xextsym =
5 4 4 5 6 6 5
2 1 1 2 3 3 2
2 1 1 2 3 3 2
5 4 4 5 6 6 5
5 4 4 5 6 6 5
2 1 1 2 3 3 2
```

2.11.2 一维延拓——图形接口工具

1. 启动一维图形延拓工具并装载信号

在图 2-4 所示的小波工具箱主窗口中单击【Signal Extension】按钮。在出现的窗口主菜单里选取【File】→【Load Signal】菜单命令，选择 MATLAB 安装目录下的 toolbox/wavelet/wavedemo 子目录中的 noisbloc.mat 文件。如图 2-99 所示。

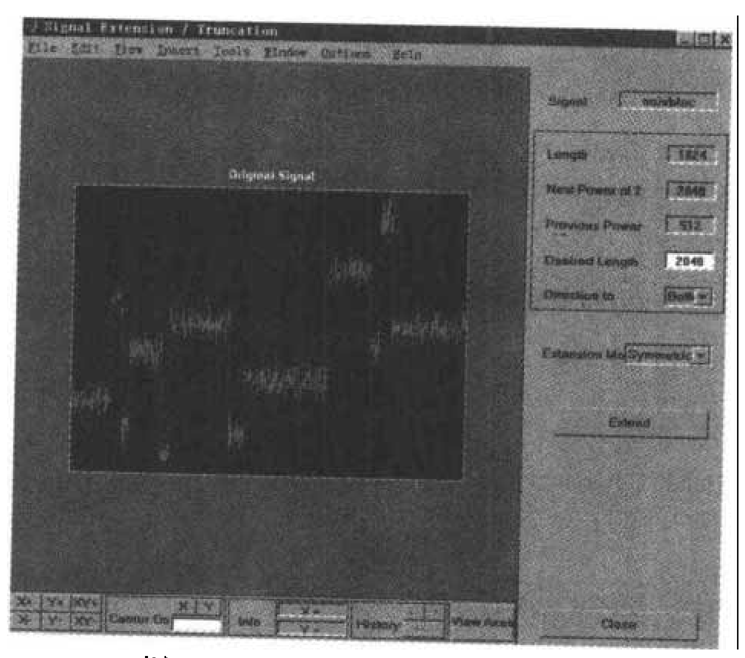


图 2-99 noisbloc.mat 文件运行结果

2. 延拓信号

在图 2-99 右边的 Desired Length（期望延拓长度）框里输入 1300，在 Direction to（延拓方向）下拉框里选择 Left（即向左延拓）。Extension Mo（延拓模式）下拉框选择系统默认的 Symmetric（对称性延拓）。然后单击【Extend】按钮，结果如图 2-100 所示。可以看出，图中红色矩形框里是原始信号，黄色矩形框里是延拓后的信号。很明显，延拓是通过对称复制原始信号靠左边的一部分数据来进行的。

再重新选择 Direction to 为 Both，Extension Mo 为 Continuous。单击【Extend】按钮，结果如图 2-101 所示，看看图 2-101 与图 2-100 的区别。

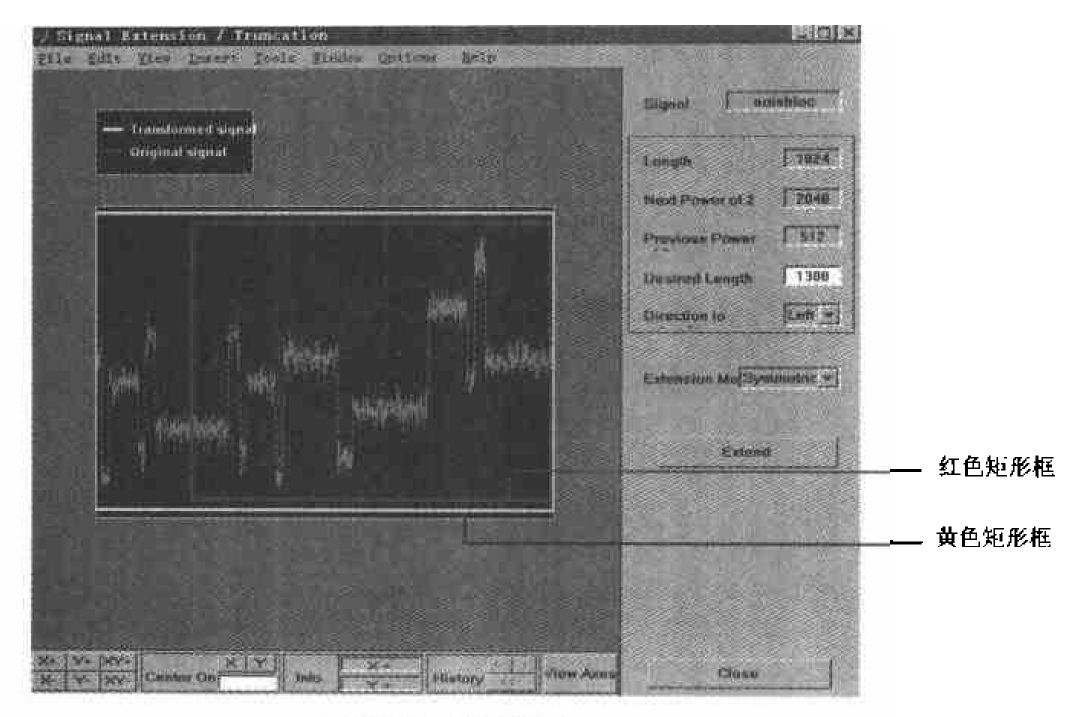


图 2-100 延拓信号

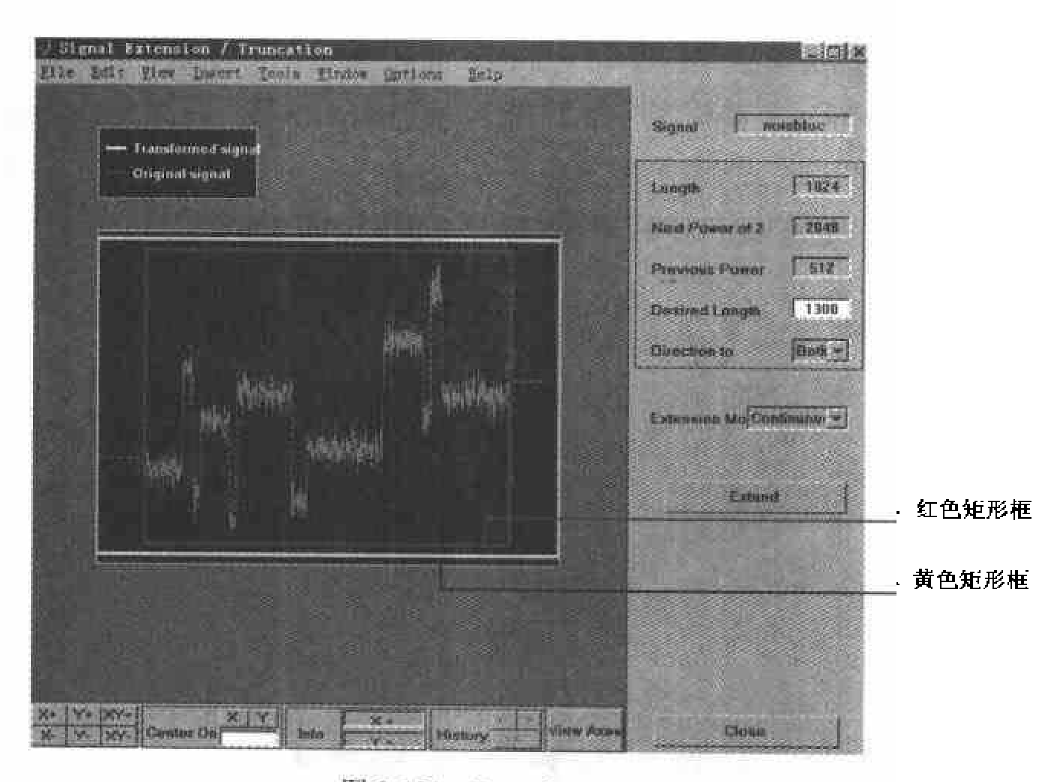


图 2-101 Extend

3. 面向平稳小波变换 SWT 的延拓

在前面我们曾提到，对信号进行在第 k 层进行平稳小波变换 SWT，2^k 必须被信号长度整除。这里一维信号延拓工具提供了用于 SWT 的延拓模式。

在图 2-101 所示的 Extension Mo 里选择 For SWT，然后单击【Extend】按钮，出现图 2-102 所示结果。

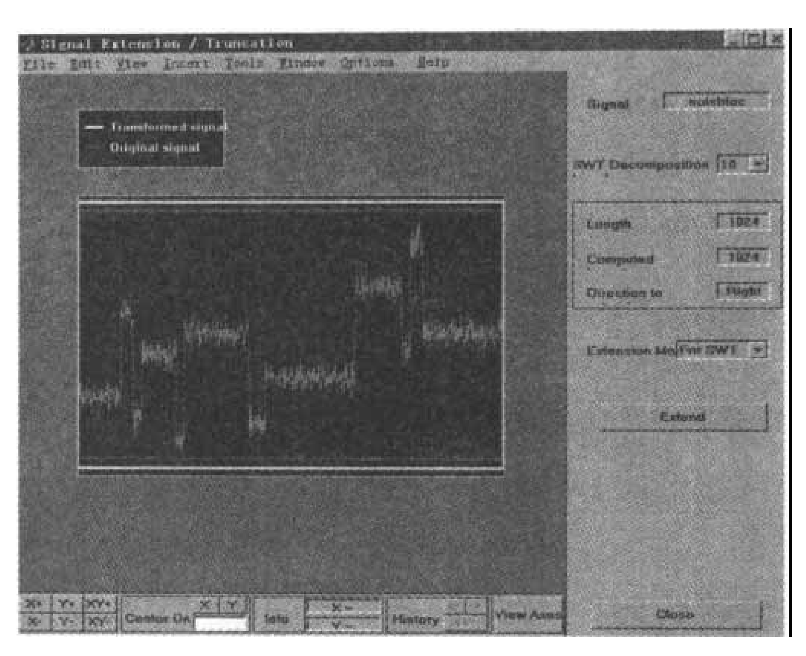


图 2-102 面向平稳小波变换 SWT 的延拓

可以看出，由于信号原始长度就是 1024＝2^{10}，所以并没有进行延拓。

4．系统自带例子演示

从图 2-102 所示的主菜单里选择【File】→【Demo Extension】命令，选取最后一个例子：freqbrk－extension－mode:swt－level:10－direction:right，如图 2-103 所示。

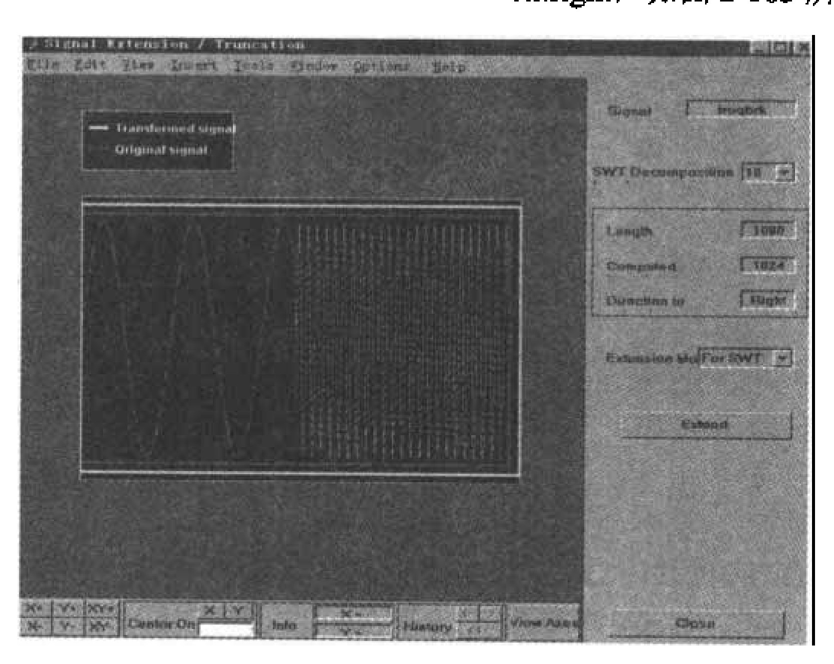


图 2-103 系统自带例子演示

由于原始信号长度为 1000，进行 SWT 的层次是 10，所以延拓工具向右进行了微小的周期延拓。延拓后的信号长度为 1024。

在图 2-103 中重新选择 SWT Decomposition 为 4，然后单击【Extend】按钮，则延拓工具将初始信号向右进行了微小的周期性延拓，使其长度达到 1008。一维 1008 是大于 1000 而又能被 6（2^4）整除的最小整数。

5. 信号截断

选择 Extension Mo 为 Periodic，在期望延拓长度编辑框 Desired Length 里输入 900 并回车，然后单击【Truncate】按钮，截断处理如图 2-104 所示。

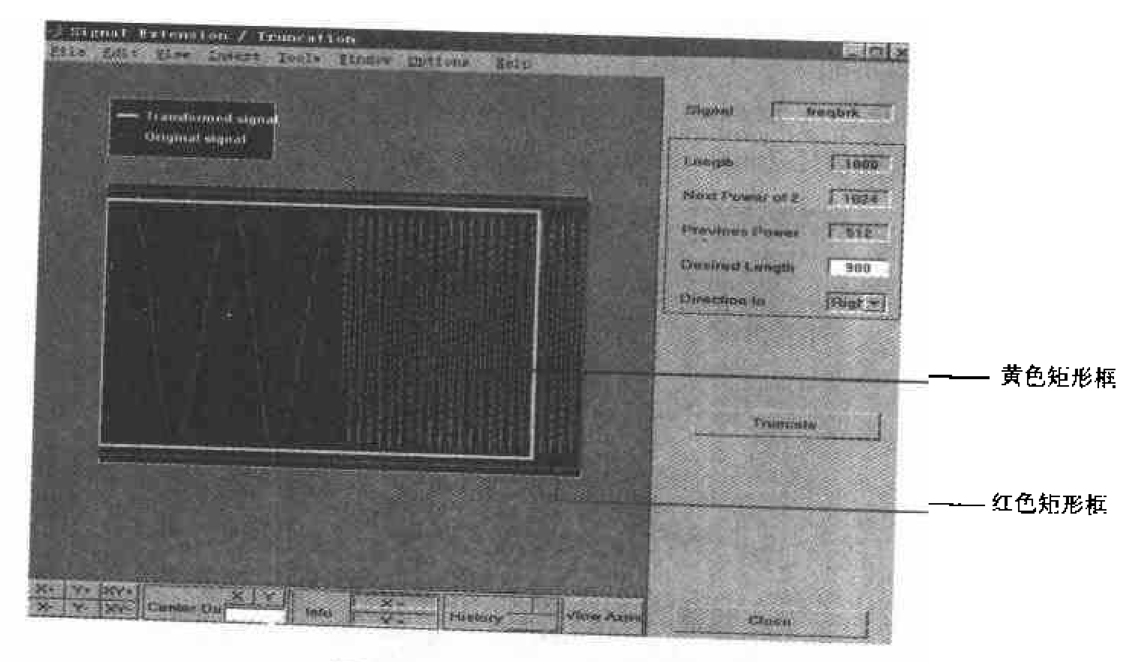


图 2-104 信号截断

可以看出，红色矩形框里的原始信号靠右边的 100 个值被截断，黄色矩形框里就是截断处理后的信号。

2.11.3 延拓或截断后的信号保存

在图 2-104 所示的主菜单中选取【File】→【Save Transformed Signal】命令，以 tfrqbrk.mat 保存在当前目录，然后将其装载进工作空间：

```
>>load tfrqbrk
whos
  Name        Size       Bytes    Class
  tfrqbrk     1×900      7200     double array
```

2.12 二维信号延拓或截断

这一节讲述如何利用小波工具箱进行二维延拓或截断。

2.12.1 二维延拓——命令行方式

小波工具箱函数中用于二维图像信号延拓的函数也是 wextend，用法可参见上一节的详细说明和例子。

2.12.2 二维延拓——图形接口方式

首先启动二维图形延拓工具并装载信号

在图 2-4 所示的小波工具箱主窗口中单击【Image Extension】按钮。在出现的窗口主菜单里选取【File】→【Demo Extension】命令，选择第三个演示例子：wbarb－extension－mode:sym－size:[512 200]－direction:[right both]，如图 2-105 所示。

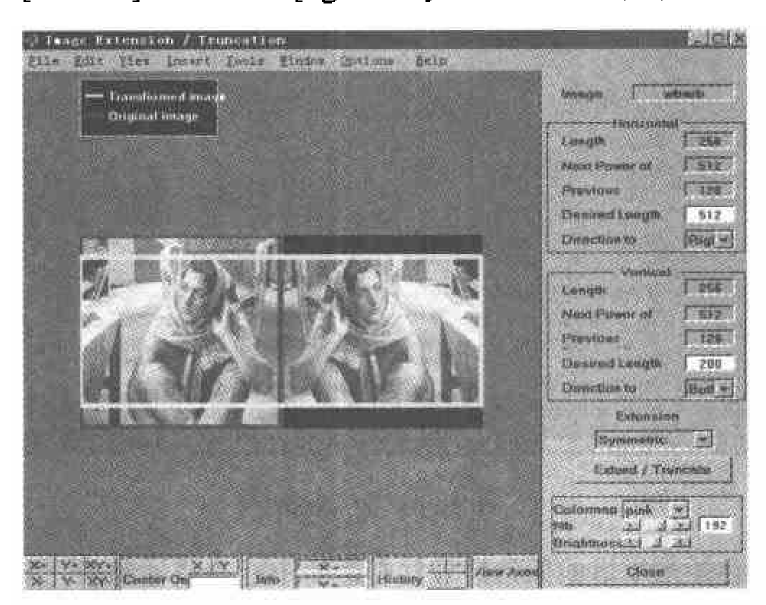


图 2-105 二维图形延拓工具

可以看出，原始图像被对称延拓。在图 2-105 窗口右边可以分别设置水平和垂直方向延拓或截断的参数，含义和上一节一维延拓一样。用户也可以通过命令 wavedemo 来查看更多的小波工具箱演示例子。

2.12.3 延拓或截断后的图像保存

在图 2-105 所示的主菜单中选取【File】→【Save Transformed Image】命令，以 ewbarb.mat 保存在当前目录。

第 3 章　MATLAB 6.5 小波包分析示例

前面已经提到，小波包相对于小波的主要优点是小波包可以对信号的高频部分做更加细致的刻画，对信号的分析能力更强，因此掌握 MATLAB 6.5 小波包分析的方法也很重要。MATLAB 6.5 的小波包工具箱可以以命令窗口和图形工具两种方式完成。本章将进一步引导读者学会使用一维和二维小波包图形工具。小波包树结构中也有一些面向对象的设计特征，可参见第 4 章面向对象的设计。

本章主要内容：

- 小波包分析简介
- 一维小波包分析
- 二维小波包分析
- 图形工具中的数据交换

3.1　MATLAB 6.5 小波包分析简介

小波包分解与小波分解不同的地方在于小波包分解不仅对低频部分进行分解，而且还对高频部分也进行分解。例如应用小波包进行压缩和消噪包括下面四步：

1．分解

对信号 X 采用给定的小波进行 N 层小波包分解。

2．计算最佳（优）树

根据给定的熵标准计算初始树的最佳子树。在小波包图形工具中通过单击【Best Tree】按钮可以快速而容易地完成上述计算。

3．对小波包系数进行阈值处理

对每一个小波包系数（除低频部分），选择一个阈值对其进行阈值处理。小波包图形工具自动给出一个初始阈值，通常在大多数情况下该阈值是较合理的。然而，有时候用户必须自己通过试验重新选择阈值以满足特定的分析和设计标准。图形工具可以方便地实现这一过程。

4．重构

在初始信号第 N 层低频系数和经过阈值处理后的高频系数的基础上进行小波包重构。

在下一节中，我们将举一个采用一维小波包图形工具进行信号压缩和消噪的例子。

3.2 一维小波包分析——图形接口方式

由于小波包分解树结构内在的复杂性，这里我们首先通过图形工具来获得小波包分析的感性认识，在后面的小节中将讲解如何在命令窗口方式中采用小波包分析函数进行小波包分析。

3.2.1 采用一维小波包进行信号压缩

1. 启动 Wavelet Packet 1-D 工具

在 MATLAB 命令符下键入 wavemenu，出现如上一章图 2-4 所示的小波工具箱主菜单窗口（Wavelet Toolbox Main menu）。选择 Wavelet Packet 1-D，出现如图 3-1 所示的一维小波包图形工具。

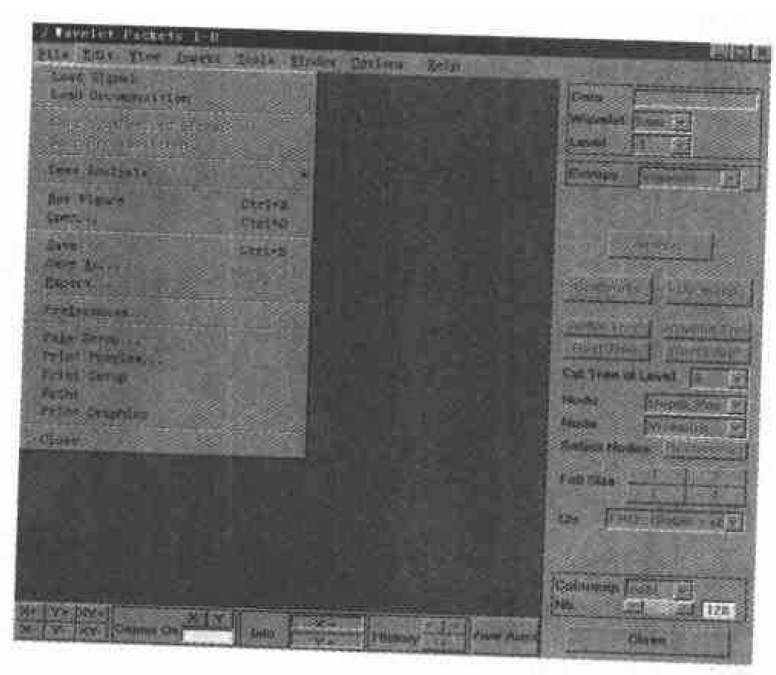


图 3-1 一维小波包图形工具

2. 装载信号

在图 3-1 所示主菜单中单击【File】→【Load Signal】菜单命令，选择 MATLAB 6.5 安装目录下的 toolbox/wavelet/wavedemo 子目录中的 sumlichr.mat 文件，如图 3-2 所示。

可以看出，一维小波包图形工具窗口与前面我们用过的一维小波图形工具类似，左边是信号的显示区域，右边是对信号进行小波包分析的各种按钮和参数选择设置框。在左边的显示区域有 4 个图形区：

（1）Analyzed Signal 在右上角，显示原始信号波形。

（2）Decomposition Tree 在左上角，显示小波包分解树结构。右移上端的滚动条可以放大图形，左移滚动条则缩小图形；右移下端的滚动条可以将图形往右移，左移滚动条则可以将图形往左移。当对信号进行多层小波包分解后的树结构很复杂时，用户可以通过这



两个滚动条方便地查看各个节点的细节。

（3）Node Action Result 在左下角，显示某个节点分析的结果。

（4）Colored Coefficients for Terminal Nodes 在右下角，显示信号小波包分解后系数的灰度图像。在一维小波包图形工具中，对一个信号是按照完整二叉树的方式来进行 n 层的小波包分解，这样终节点有 2^n 个。图形工具对第 n 层的 2^n 组分解系数以灰度图像的方式来显示其结果。在其下方还有一个标为 Scale of Colors from Min to Max 的颜色条，用来表示系数大小与颜色的对应关系。

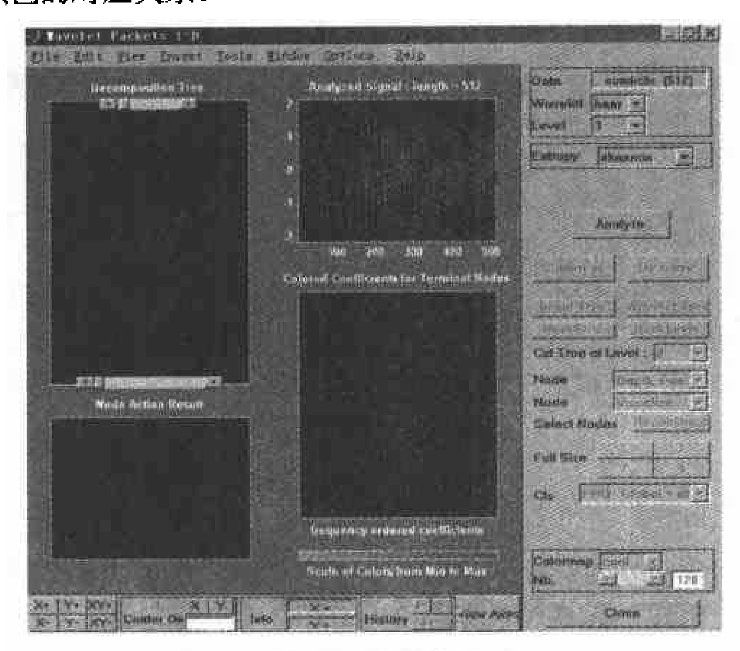


图 3-2　装载信号

3．**分析信号**

在图 3-2 窗口的右边可以对信号分析进行合理的设置。小波包基函数 Wavelet 选择框选取 db2，分解层次 Level 取 4，Entropy（熵标准）选择框选取 threshold。然后在出现的阈值参数框里输入 1，单击【Analyze】按钮，该按钮执行的功能是对信号进行小波包分解并将分解结果显示在窗口中，如图 3-3 所示。

这里对 Entropy（熵标准）选择框进行一下解释。可用的阈值熵标准类型有（这里设 E 代表熵，s 代表信号，s_i 代表信号 s 在一个正交小波包基上的投影系数）：

（1）Shannon 熵：$E(s_i)=-s_i^2\log s_i^2$，$E(s)=-\sum s_i^2\log s_i^2$（约定 0log0＝0）。

（2）threshold 阈值熵：设阈值为 ε，如果$|s_i|>\varepsilon$，则 $E(s_i)=1$，否则 $E(s_i)=0$。所以此时 $E(s)$等于信号所有采样值大于阈值 ε 的采样点个数。

（3）nor 范数熵。

（4）log energe 对数能量熵：$E(s_i)=\log s_i^2$，$E(s)=-\sum\log s_i^2$。

（5）sure 熵：类似阈值熵，只不过这里阈值 $\varepsilon=\sqrt{2\log_e(n\log_2(n))}$，$n$ 为信号长度。

（6）user 用户熵：用户以 M 文件格式自己定义熵。

关于熵更详细的信息可以利用系统的帮助功能来查看用来计算小波包的熵的函数

wentropy 的用法。

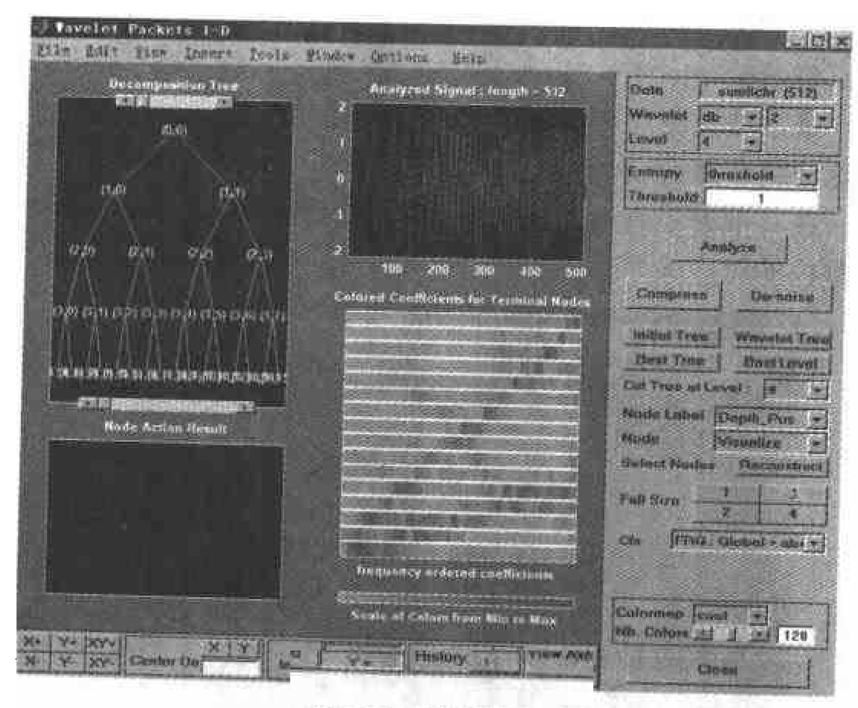
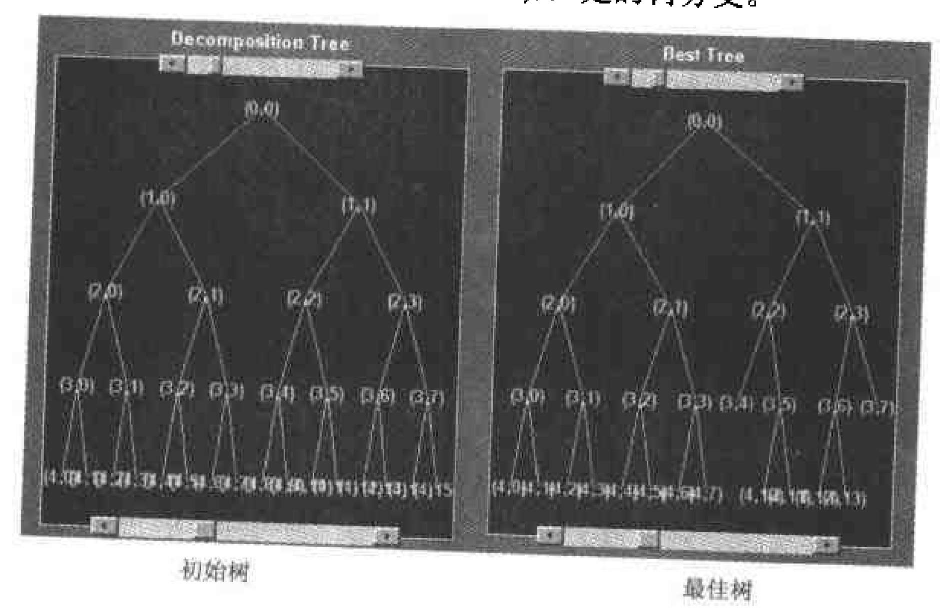


图 3-3　分析信号

4. 计算最佳树

由于从小波包分解树中重构原始信号有许多可选的方法，所以在压缩信号之前我们先根据上面的熵标准计算出初始树的最佳子树。在图 3-3 中的右上方单击【Best Tree】按钮，其功能就是在执行信号的小波包分解之后用 Entropy 选择框确定的熵标准计算出最佳的小波包分解树。在它上方还有一个【Initial Tree】按钮，其作用是进行一个信号的小波包分解的初始树并显示。用户可以利用这两个按钮对初始树和最佳树进行对比，如图 3-4 所示。

可以看出，最佳树去掉了节点（3,4）和（3,7）处的树分支。



初始树　　　　最佳树

图 3-4　计算最佳树



5．选择阈值进行信号压缩

图 3-3 右上方的【Compress】按钮功能是在信号小波包分解后对各层的高频系数进行阈值量化处理，然后再根据处理后的系数进行信号重构，从而达到对信号用小波包方法进行压缩的目的。进行阈值量化时，熵标准由 Entropy 选择框确定。注意，在一维小波包图形工具中，只有全局阈值。单击【Compress】按钮，出现如图 3-5 所示的一维小波包压缩窗口（彩色效果图见彩插 4）。

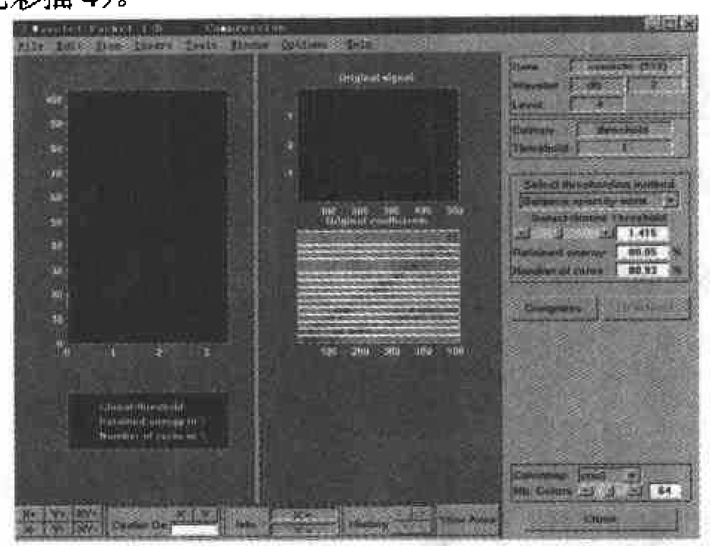
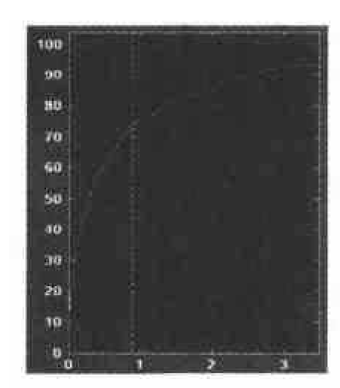


图 3-5　一维小波包压缩窗口

图 3-5 垂直方向的黄点线代表系统自动选取的一个阈值（1.482），这个值是在置零系数的百分比（图中随阈值增加而增加的蓝色曲线）和压缩后保留的信号能量百分比（图中随阈值增加而减小的紫色曲线）之间进行折中得到的。这意味着小波包分解后的所有小于 1.482 的高频系数将被舍弃。

从图 3-5 可以看出，在阈值为 1.482 的情况下，信号压缩后仅保留了 81.49%的能量，这在信号是含有尖峰的振荡信号时可能是不可接受的，因为这样误差太大。这时用户可以在上面窗口右方的阈值控制框里选择新的阈值以保留更多的能量。

在阈值框里输入新的阈值 0.8938（这个值是通过多次尝试得到的针对该例的最佳阈值）并回车，变化如图 3-6 所示。可以看出，当我们将阈值由 1.482 降到 0.8938 后，代表阈值的黄点线左移，压缩后保留的能量从 81.49%增加到 90.81%，置零的系数百分比（等于压缩量）由 81.55%降到 74.07%。



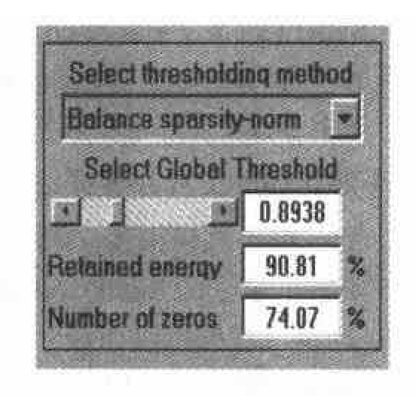


图 3-6　置零系数百分比和能量保留百分比

另外，用户还可以直接在图 3-5 中代表阈值的黄点线上按住鼠标左键不放拖动该线来更直观地调整阈值，来观察置零系数百分比和能量保留百分比的变化情况。

6．信号压缩

单击图 3-5 窗口中右边的【Compress】按钮，则一维小波包图形工具采用我们先前选取的阈值标准对信号进行压缩处理，结果如图 3-7 所示（彩色效果图见彩插 5）。

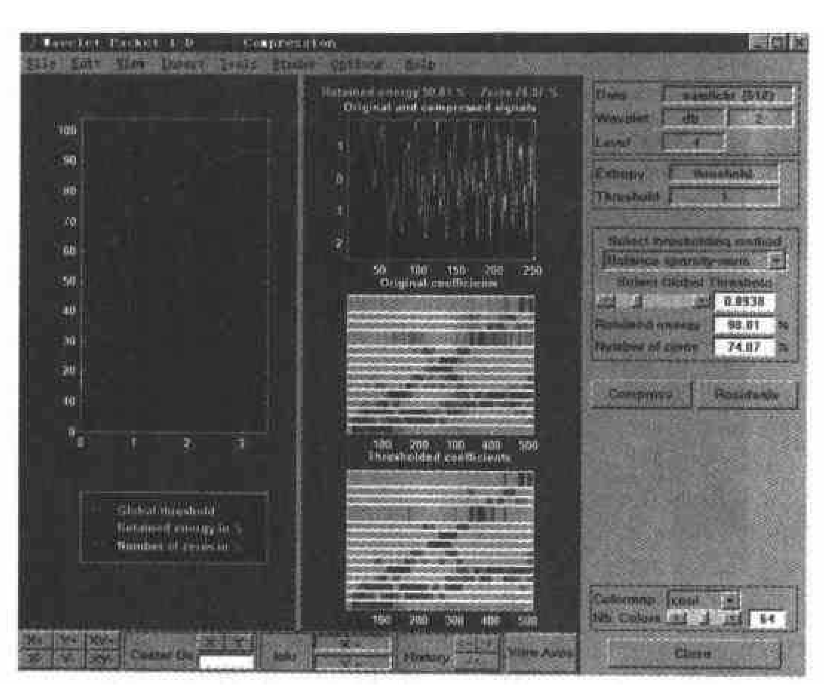


图 3-7　信号压缩

可以看出，压缩信号（黄色）和原始信号（红色）在同一个显示框中，并且经过阈值量化处理后的系数和原始系数也在下面的显示区对比显示出来。

如果采用相同的参数对上面同一个信号进行小波压缩，将只有大约 88％的能量被保留，置零系数的百分比也只有大约 58％，这体现了小波包压缩的优越性。用户可以回到图 3-3 中单击【Wavelet Tree】按钮后再进行压缩来验证。

3.2.2　采用一维小波包进行信号消噪

这一小节我们采用一维小波包图形工具来对一个被噪声污染的线性调频信号进行消噪。这里我们采用 SURE(Stein's Unbiased Estimate of Risk)熵标准来选择阈值进行消噪，因为当分析信号满足下面的模式时采用这种熵标准消噪效果最好。

$x(t)=f(t)+e(t)$，这里 $e(t)$为具有零均值和单位方差的高斯白噪声。

1．启动一维小波包图形工具并装载待分析信号

在图 3-1 所示的一维小波包图形工具主菜单中单击【File】→【Load Signal】命令，选择 MATLAB 6.5 安装目录下的 toolbox/wavelet/wavedemo 子目录中的 noischir.mat 文件。该原始信号长度为 1024，我们可以在命令窗口输入表达式，计算出采用 SURE 熵标准时的阈值为 sqrt(2.*log(1024.*log2(1024)))＝4.2975。

2. **分析信号**

在小波包基函数 Wavelet 选择框选取 db2，分解层次 Level 取 4，Entropy（熵标准）选择框选取 sure，然后在出现的阈值参数框里输入 4.2975，再单击【Analyze】按钮对信号进行小波包分解，结果显示如图 3-8 所示。

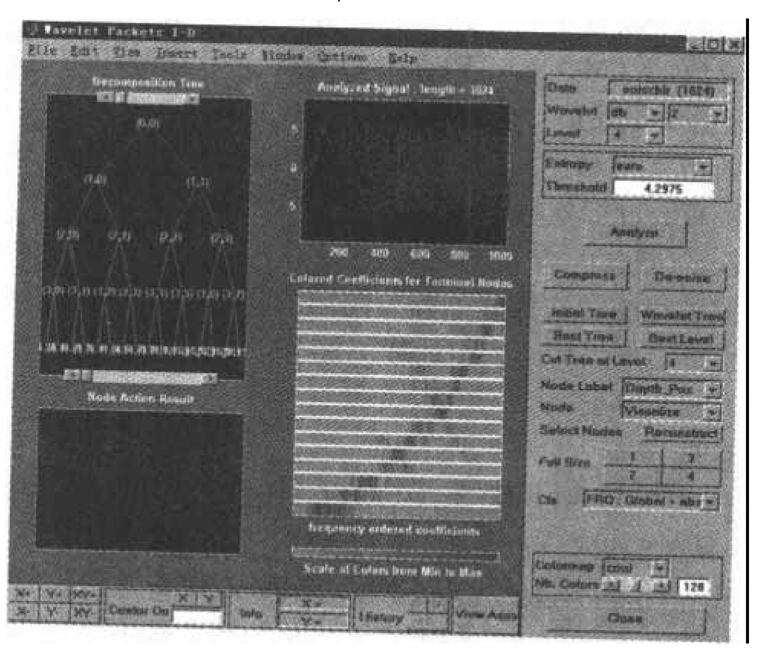
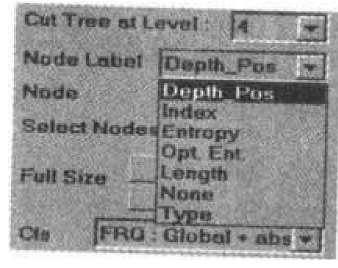
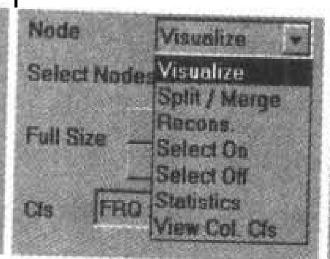


图 3-8 分析信号

注意到图 3-8 窗口右边有些选择框，我们将其放大如图 3-9 所示，并对其意义逐一解释。





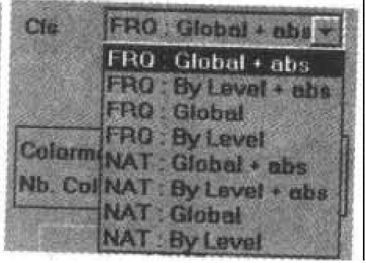


图 3-9 选择框

（1）Cut Tree at Level 选择框

此项用来将小波包分解树剪切到某一层。如果对某个信号进行 n 层分解，该选择框里的数字是从 0 到 n。这样如果该信号只需要分解到第 k 层就可以满足分析要求，那么就可以在该选择框里输入数字 k。

（2）Node Label 选择框

此项用来选择节点标志方式，有 7 种表示方式：Depth-Pos（深度–位置）方式采用深度位置的二维数组进行标注，如（1,0）、(2,1）等；Index（索引）方式以节点的索引值来标志节点，根节点索引值为 0，然后从上到下、从左到右依次为 1,2,3…；Entropy（熵值）方式在各个节点处标志其对应的熵值；Opt.Ent（最优熵）方式根据该节点是否是最优熵来标志，如果不是则标为“NaN”（Not a number）；Length 方式标志各节点的数据长度；None 是无标志方式，即什么也不标注；Type（类型）方式根据各节点是高频还是低频相应标注 d 或 a。

（3）Node 选择框

此项用来对节点操作进行选择，有 7 种方式：Visualize（显示）方式时，单击左边树结构图中的任意一节点，则在左边的 Node Action Result 显示框中就会显示该节点的系数曲线；Split/Merge（分解/合成）方式时，单击树结构中的任一节点，如果该节点是终节点，则对该节点进行下一层的小波包单层二叉分解（Split），如果该节点不是终节点，就将该节点以下的所有子节点剪切（Merge）；Recons.（重构）方式时，对所选节点的系数进行重构，使重构后的长度等于原始信号长度；Select On（选中）方式时，用鼠标选中树结构中的几个节点（选中的节点呈绿色），单击出现在该选择框下的【Reconstruct】按钮，可以对这几个节点的分解系数进行重构，完成后再选择 Select Off（取消选中）来取消选中的这几个节点；Statistics（统计）方式时，单击树结构中的任一节点会弹出一个统计窗口，将所选节点的数学特征进行统计计算并以直方图形式显示出来；View Col.Cfs（查看终节点系数值的颜色）方式时，单击树结构中的任一终节点，则在左边图中显示该节点系数值对应的灰度图像。

（4）Cfs 系数染色选择框

从图 3-9 可以看出，染色方式总的分为 FRQ 和 NAT 两类，FRQ 表示系数值以频率分布来度量，NAT 表示系数值以奈特（信息量的自然单位）来度量。在每种度量方式下又都有以下四种染色方式：Global+abs（全局＋绝对值）方式将原始信号和所有小波包分解层的系数值、绝对值的范围作为颜色的染色范围。总的颜色数由 Nb.Colors 滚动条代表的值来决定，此时用户可以很清楚地看到所有终节点和原始信号的数值分布情况；By Level+abs（层次＋绝对值）方式下，每个终节点的系数和原始信号绝对值的染色是独立的，即每一层都有相同的颜色数，此时可以方便地查看各层系数值的分布情况。Global（全局）和 By Level（按层次）方式和上面两种方式的染色类似，只是染色的依据不是绝对值，而是本身的大小。

3．计算最佳树并完成消噪

首先单击【Best Tree】按钮，计算出最佳树以使下面的消噪处理更有效。然后再单击【De-noise】按钮，出现如图 3-10 所示的一维小波包消噪窗口。

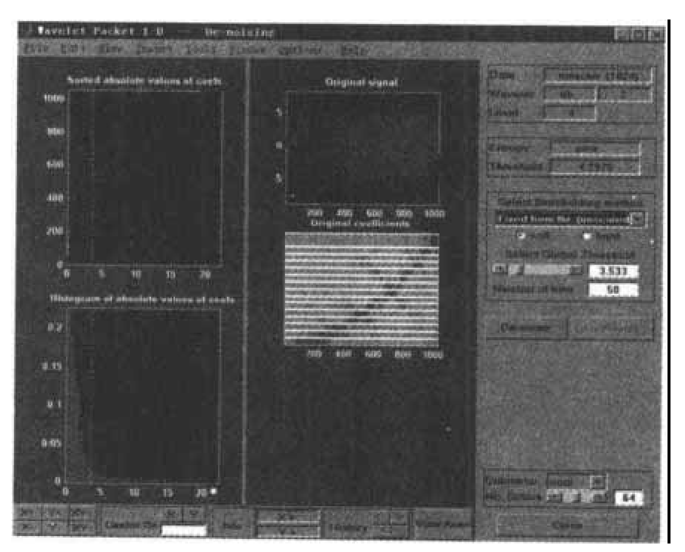


图 3-10　一维小波包消噪窗口

单击图 3-10 中的【De-noise】消噪按钮，消噪结果如图 3-11 所示（彩色效果图见彩插 6）。

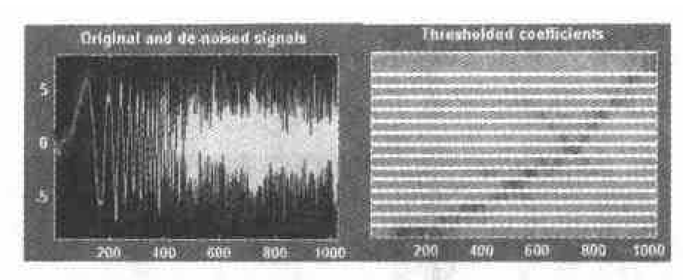


图 3-11 消噪结果

图 3-11 中红色的是原始信号，黄色的是消噪后的信号。

最后要说明的是，用户也可采用小波工具箱函数 wpdencmp 来完成上面的消噪或压缩过程。

3.3 二维小波包分析——图形接口方式

这一节我们将利用二维小波包图形工具来分析和压缩一幅指纹图像。这具有实际意义，例如，美国联邦调查局有一个庞大的指纹库，里面大约有 3 千万个指纹，将其以电子形式进行保存要花费大量的财力。而通过小波进行压缩，可以达到 15：1 的压缩比，而且效果比传统的 JPEG 压缩格式要好。

值得一提的是，国际标准图像格式 JPEG 2000 将把小波应用在压缩和量化过程中，这体现了小波是一种强有力的压缩工具。

3.3.1 启动 Wavelet Packet 2-D 工具

在 MATLAB 命令符下键入 wavemenu，出现如上一章图 2-4 所示的小波工具箱主菜单窗口（Wavelet Toolbox Main menu）。选择 Wavelet Packet 2-D，出现如图 3-12 所示的二维小波包图形工具。

可以看出，图 3-12 与一维小波包图形工具很相似，只不过处理的是图像。

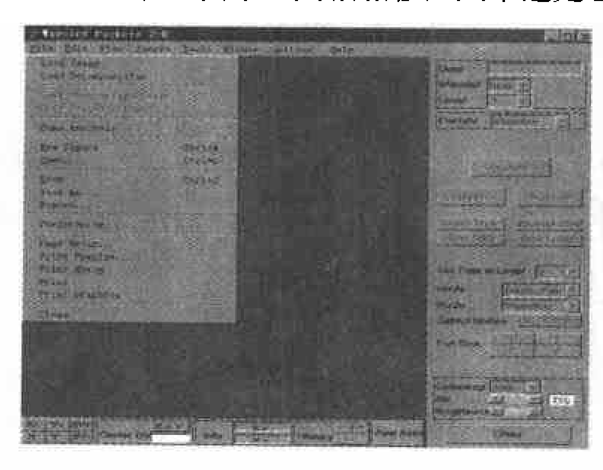


图 3-12 二维小波包图形工具

3.3.2 装载信号

在图 3-12 所示的主菜单中单击【File】→【Load Image】菜单命令，选择 MATLAB 6.5 安装目录下的 toolbox/wavelet/wavedemo 子目录中的 detfingr.mat 文件，如图 3-13 所示。

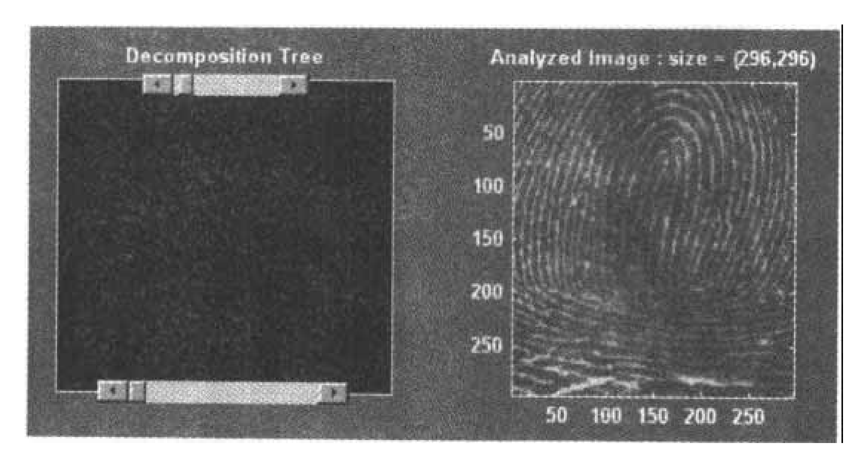


图 3-13 装载信号

3.3.3 分析图像

在图 3-13 窗口的右边对信号分析进行合理的设置。小波包基函数 Wavelet 选择框选取 haar，分解层次 Level 取 3，Entropy（熵标准）选择框选取 shannon，然后单击【Analyze】按钮。分解结果如图 3-14 所示。

可以看出，图 3-14 所示窗口右边是对图像进行分析的各种选择框按钮，和一维小波包图形工具中的含义一样，这里不再赘述。

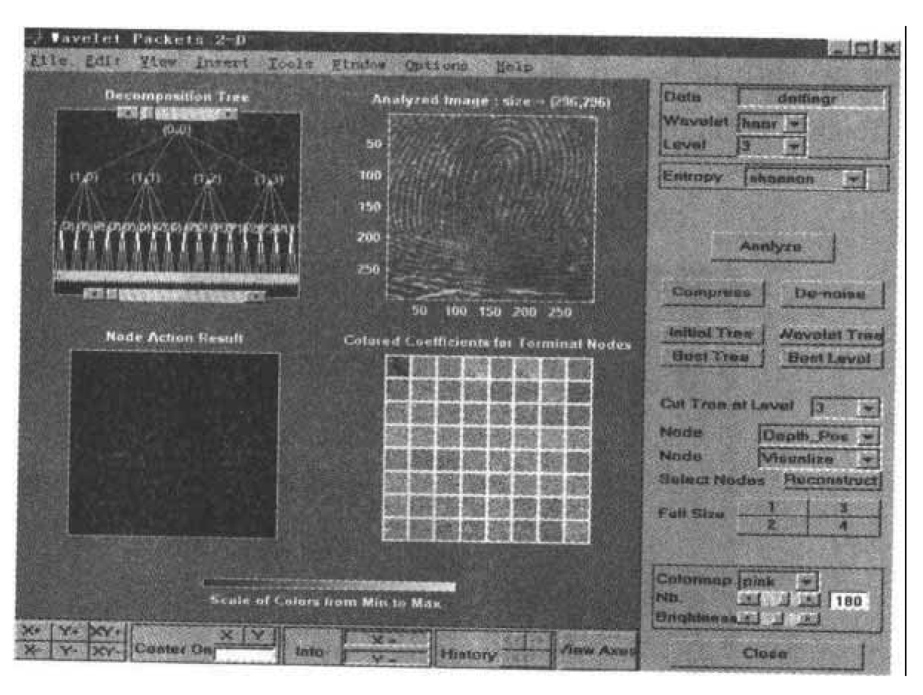


图 3-14 分析图像



3.3.4 计算最佳树

压缩指纹图像之前我们先按熵标准计算出初始树的最佳子树。在图 3-14 中单击【Best Tree】按钮。读者可以发现最佳树比初始树减少了相当一部分分支。

3.3.5 利用小波包压缩图像

在图 3-14 中单击【Compress】按钮，出现图 3-15 所示的二维小波包压缩窗口。

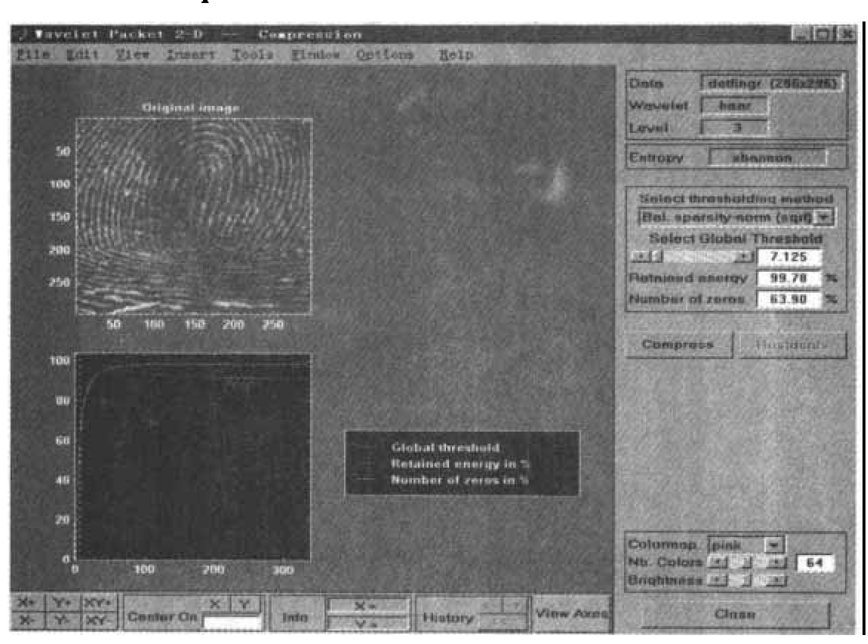


图 3-15 二维小波包压缩窗口

可以看出，按照系统给出的默认阈值 7.125，在保留了 99.78%的能量前提下压缩比约为 36%。读者还可以通过试验找出更佳的阈值，使得在图像不失真的前提下压缩比达到最大。注意，我们这里还没有对图像进行量化，只是仅仅将某些分解小波包系数置零。这可以看做一个更广泛的压缩过程中的预压缩步骤。

在图 3-15 中的阈值编辑框里输入 30 并回车，可以发现有将近 92%的系数被置零，但仍保留了将近 98%的原始图像能量，获得了更好的压缩效果。读者可以在这里采用小波进行图像压缩进行对比，说明小波包压缩比小波压缩更有优越性。

在上面改变阈值后单击【Compress】按钮开始压缩图像，图 3-16 是压缩前后的图像对比。

最后单击图 3-15 窗口下面的【Close】按钮，会出现更新合成图像（Update the syethesized image）对话框，单击【Yes】按钮结束压缩过程回到二维小波包图形工具主窗口。

此时读者可以尝试自己定义压缩策略，即改变阈值、熵标准和小波包基函数，看看能不能获得更好的压缩比。试验证明，bior6.8 比 haar 更适合对这幅指纹图像进行压缩，可以获得更高的压缩比。

在本节结束之前，我们仔细分析一下图 3-14 中右下方的 Colored Coeffiients for Terminal

Nodes（终节点系数染色）图，考虑针对该幅图像进行最佳树分解，如图 3-17 所示。

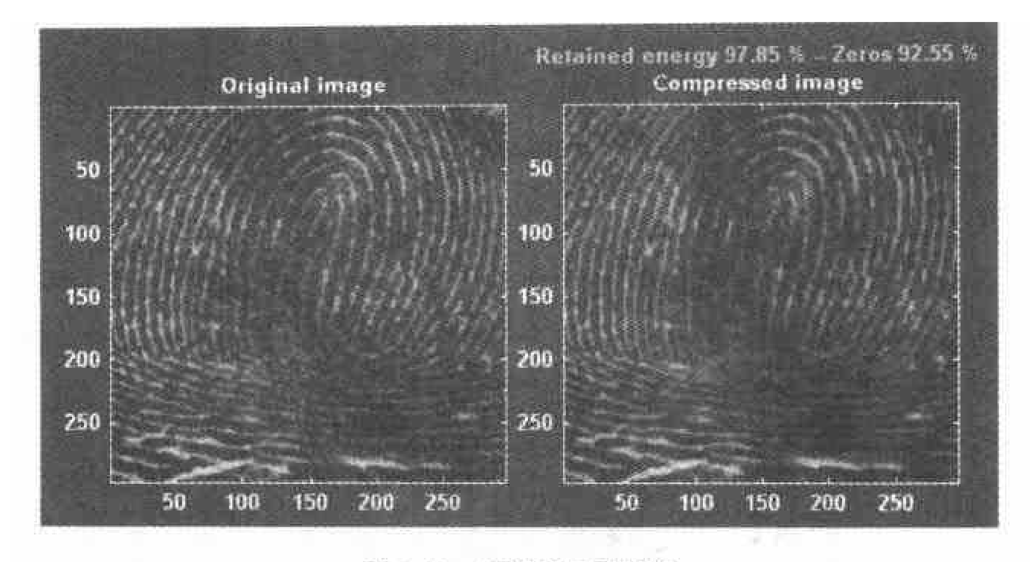


图 3-16　压缩前后的图像

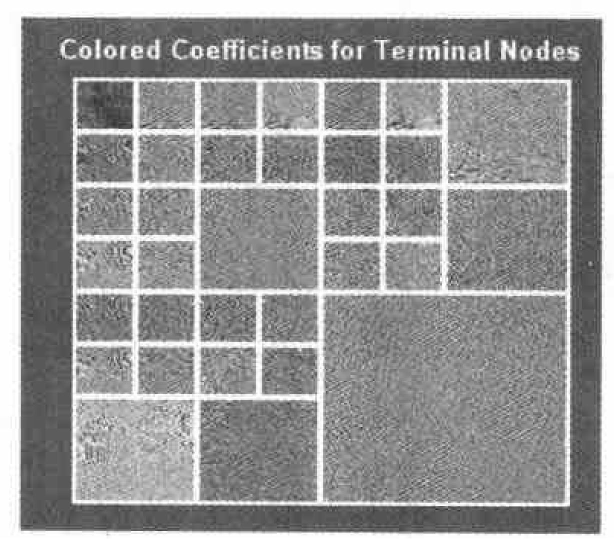


图 3-17　终节点系数染色

图 3-17 显示了哪些高频部分被分解或没被分解。大的正方形表示该部分没有像小的正方形一样进行多层次的分解。考虑图 3-18 所示的一个 2 层小波包分解。

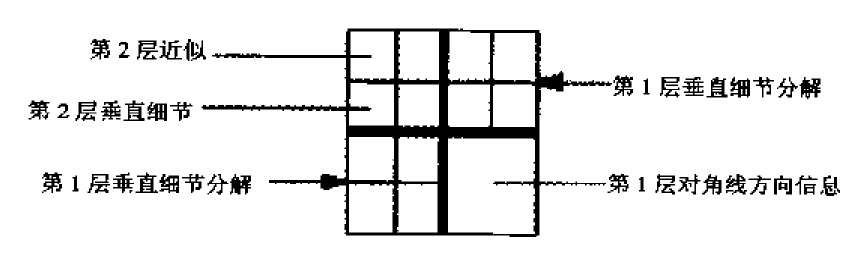


图 3-18　2 层小波包分解

可以看出，此情形下最佳树算法显然是保留对角线方向的高频信号不再进行分解。为什么是这样的呢？大家可以注意到，原始指纹图像的主要特征集中在沿水平和垂直方向分布的变化较大的边缘上，所以我们进行最佳树计算时，选择不再对对角线方向的高频部分

进行分解，因为它们并没有提供多少原始图像信息。

3.4　图形工具中的数据交互

采用约定的格式，小波包分析图形工具可以保存分解或合成后的信号或图像数据，也可以装入小波包分解后的信号或图像数据。这些可通过一维或二维小波包图形分析工具主菜单【File】中的选项来完成，其操作和小波分析图形工具中的数据 I/O 操作类似，这里不再赘述，读者可参见相关章节。

第4章 面向对象的设计及应用

在MATLAB 6.5小波工具箱里，一些面向对象的设计特征体现在小波包树结构中，这在上一章已经提到。当进行保存（Save）和装载（Load）涉及到对象时，理解本章的内容就显得很有必要。

这一章将引导读者理解工具箱里的对象，学会使用一些相关的函数，以及利用预先定义的树结构和一些面向对象的设计特色来扩展工具箱的功能。

本章主要内容：

- 小波工具箱里的对象简介
- 对象的简单应用
- 对象的进一步讨论
- 对象的高级应用

4.1 小波工具箱里的对象简介

小波工具箱定义了下面4类按层次排列的对象。

1. WTBO

该类是一个抽象类，小波工具箱里的任何对象都由一个WTBO类派生而来。

2. NTREE

该类专门用来进行树操作（节点标志、节点分解/合成等），也是一个抽象类，主要的方法有：

- nodejoin　改组节点
- nodesplt　分解节点
- wtreemgr　用来获取树和节点的信息（阶数、深度、终节点等）

实际上，wtreemgr方法并不直接应用，而是采用函数treeord、treedpth、leaves、nodeasc、get等。

3. DTREE

专门用于对树进行与向量或矩阵有关的处理，也是抽象类，其中某些方法必须重载，如split、merge、recons等。

4. WPTREE

用来管理小波包，有些方法专门针对该类，如bestlevt、wp2wtree等。

用户可以通过在命令窗口输入help wavelet，然后在Wavelets Packets Algorithms和Tree Management Utilities这两部分查看上面类，可用的方法如下：



Wavelets Packets Algorithms.

```
bestlevt    - Best level tree (wavelet packet).
besttree    - Best tree (wavelet packet).
entrupd     - Entropy update (wavelet packet).
wentropy    - Entropy(wavelet packet).
wp2wtree    - Extract wavelet tree from wavelet packet tree.
wpcoef      - Wavelet packet coefficients.
wpcutree    - Cut wavelet packet tree.
wpdec       - Wavelet packet decomposition 1-D.
wpdec2      - Wavelet packet decomposition 2-D.
wpfun       - Wavelet packet functions.
wpjoin      - Recompose wavelet packet.
wprcoef     - Reconstruct wavelet packet coefficients.
wprec       - Wavelet packet reconstruction 1-D.
wprec2      - Wavelet packet reconstruction 2-D.
wpsplt      - Split (decompose) wavelet packet.
```

Tree Management Utilities.

```
allnodes    - Tree nodes.
depo2ind    - Node depth-position to node index.
drawtree    - Draw wavelet packet decomposition tree (GUI).
dtree       - Constructor for the class DTREE.
get         - Get tree object field contents.
ind2depo    - Node index to node depth-position.
isnode      - True for existing node.
istnode     - Determine indices of terminal nodes.
leaves      - Determine terminal nodes.
nodeasc     - Node ascendants.
nodedesc    - Node descendants.
nodejoin    - Recompose node.
nodepar     - Node parent.
nodesplt    - Split (decompose) node.
noleaves    - Determine nonterminal nodes.
ntnode      - Number of terminal nodes.
ntree       - Constructor for the class NTREE.
plot        - Plot tree object.
read        - Read values in tree object fields.
readtree    - Read wavelet packet decomposition tree from a figure.
set         - Set tree object field contents.
tnodes      - Determine terminal nodes.
treedpth    - Tree depth.
treeord     - Tree order.
```

wptree　　　- Constructor for the class WPTREE.
wpviewcf　 - Plot wavelet packets colored coefficients.
write　　　 - Write values in tree object fields.
wtbo　　　　- Constructor for the class WTBO.
wtreemgr　 - NTREE object manager.

4.2　对象的简单应用

用户可以通过命令行函数或图形方式或两者联合来应用 WPTREE 对象。最常用的命令如下：

- plot、drawtree 和 readtree，用来得到和画出一个小波包树
- wjoin 和 wpsplt，用来改变一个小波包树结构
- get，read 和 write，用来读取或写入一个小波包树里的系数或信息

下面我们通过 4 个简单的例程来说明。

4.2.1　画树结构和小波包系数

例程 4-1 介绍怎样画树结构和小波包系数。

例程 4-1

```
load noisbump
x = noisbump;
t = wpdec(x,3,'db2');
%一维小波包分解
fig = plot(t);
%画树结构的图形
%在运行上面命令后出现的图形主菜单里将节点标志方式由 Depth-position（深度–位置）方式
%改换为 Index（索引）方式，然后单击索引值为 7 的节点，得到图 4-1
%在图 4-1 的主菜单里，Node Action 从 Visualize（显示）换为 Split-Merge（分解/合成）方式
%单击节点 2。由于节点 2 不是终节点，所以将其下面的所有子节点剪切，即 Merge
%得到图 4-2。
newt = plot(t,'read',fig);
%得到新的树
%上面 plot 函数的第一个参数 dummy，一般的语法是 newt = plot(DUMMY,'read',fig);这里
%DUMMY 是任何由 NTREE 类派生出的对象
newt = wpjoin(newt,3);
%通过命令行函数改变新的树并画出
fig2 = plot(newt);
%在画出的图中将节点标志方式由 Depth-position（深度–位置）方式改换为 Index（索引）方
%式，然后单击索引值为 2 的节点，得到图 4-3
wpviewcf(newt,1);
%将原始信号及以频率分布来度量的所有小波包分解层的系数值、绝对值的范围作为染色的
```

```
%范围，结果如图 4-4 所示，其中包含了所有终节点的系数。这里的 wpviewcf 是对小波包染
%色系数作图的函数，其作用和我们在第 3 章一维小波包图形工具里的 Cfs.（系数染色）选择
%框的作用一样
```

运行结果如图 4-1～图 4-4 所示。

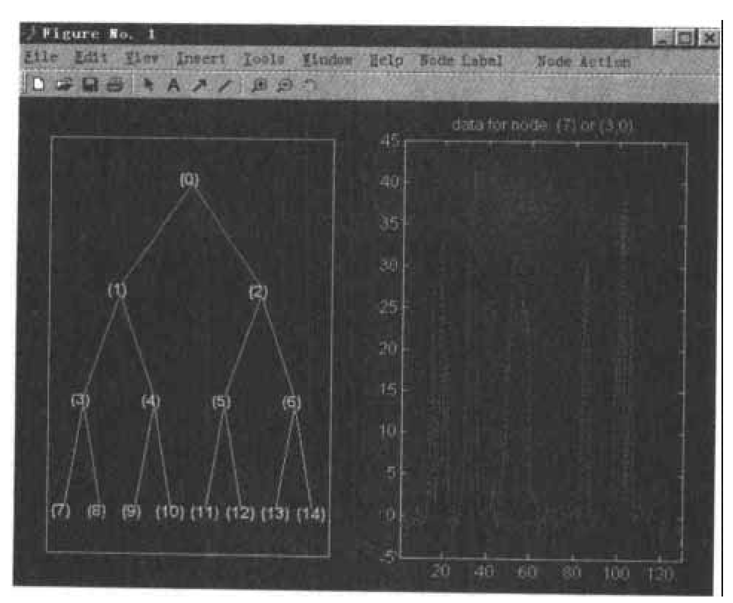


图 4-1　运行结果 1

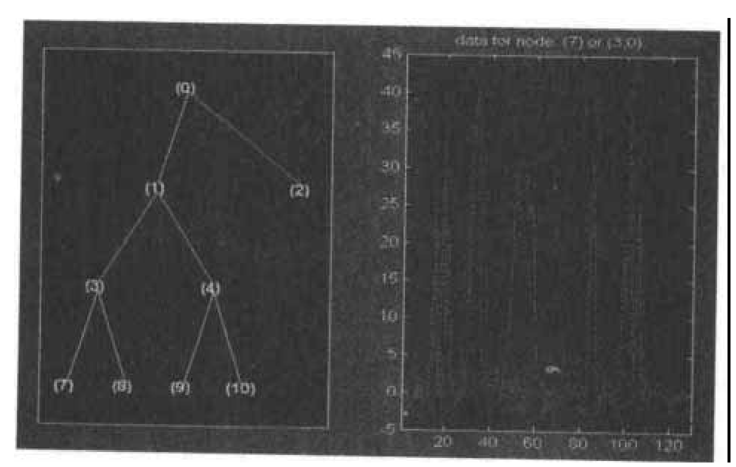


图 4-2　运行结果 2

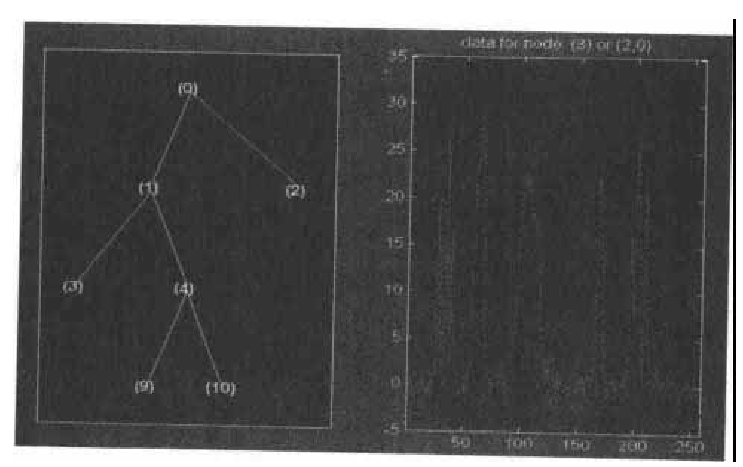


图 4-3　运行结果 3

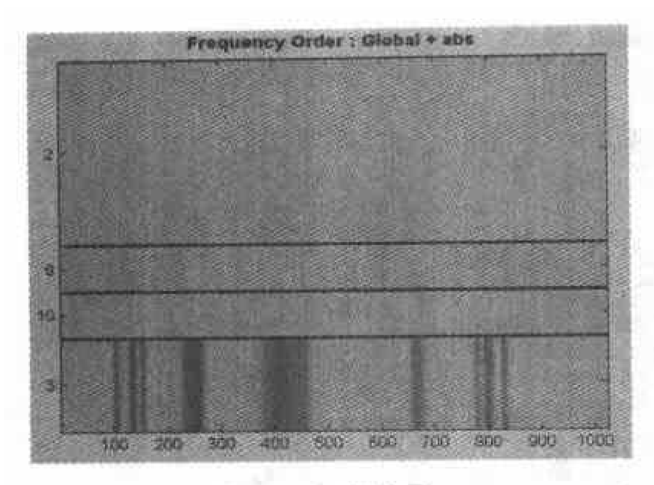


图 4-4　运行结果 4

4.2.2　drawtree 和 readtree 的使用

例程 4-2 旨在说明 drawtree 和 readtree 的使用。

例程 4-2

```
load noisbump
x = noisbump;
t = wpdec(x,3,'db2');
fig = drawtree(t);
%产生图 4-5 所示一维小波包图形工具，这里相当于采用命令行方式完成第 3 章的分析
newt = readtree(fig);
%得到新的树
newt = wpjoin(newt,3);
%重新组合小波包
drawtree(newt,fig);
%在同一个图形中显示，运行结果如图 4-6 所示。
```

运行结果如图 4-5 和图 4-6 所示。

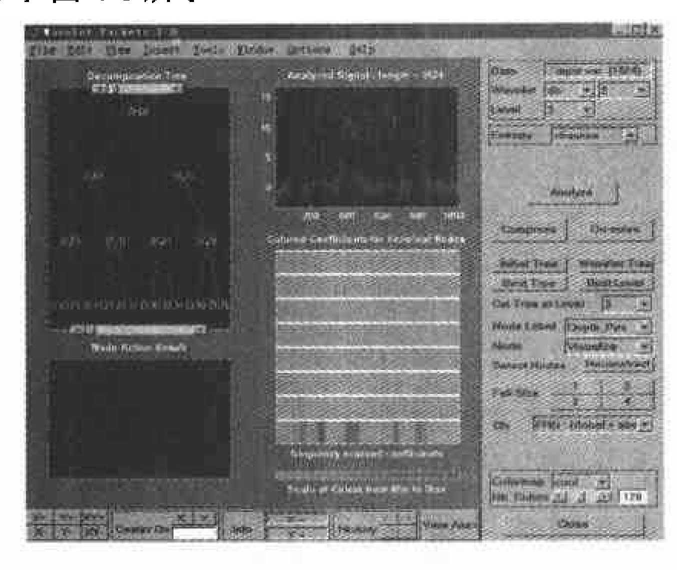


图 4-5　运行结果 1

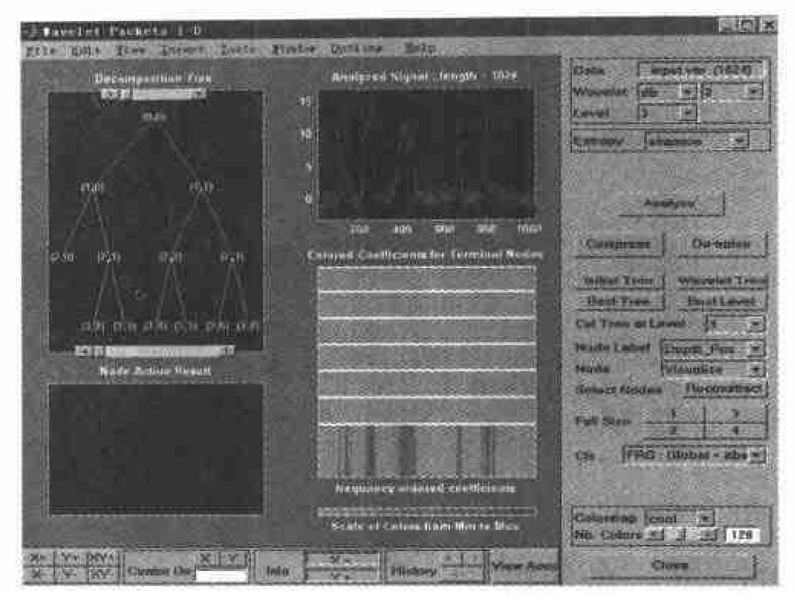


图 4-6　运行结果 2

与 WPTREE 对象有关的方法可以完成比上面更复杂的一些功能。

显然，应用方法 read 和 write，用户可以改变终节点的系数。

4.2.3　对称图形

例程 4-3 介绍对称图形。

例程 4-3

```
load gatlin2
t = wpdec2(X,1,'haar');
plot(t);
%在运行上面命令后出现的图形主菜单中将 Node Label 从 Depth-postion 切换到
%Index，并单击索引值为 0 的节点，即根节点，如图 4-7 所示
%改变四个终节点
newt = t;
NBcols = 40;
for node = 1:4
cfs = read(t,'data',node);
tmp = cfs(1:end,1:NBcols);
cfs(1:end,1:NBcols) = cfs(1:end,end-NBcols+1:end);
cfs(1:end,end-NBcols+1:end) = tmp;
newt = write(newt,'data',node,cfs);
end
plot(newt)
%在上面命令运行后出现的窗口主菜单中再将 Node Label 从 Depth-postion 切换到
%Index，并单击索引值为 0 的节点，即根节点，如图 4-8 所示
%可以看出，图 4-7 和图 4-8 的图像正好左右对称，这是因为我们通过 read 和 write
%将四个终节点的系数左右对调的原因
```

运行结果如图 4-7 和图 4-8 所示。

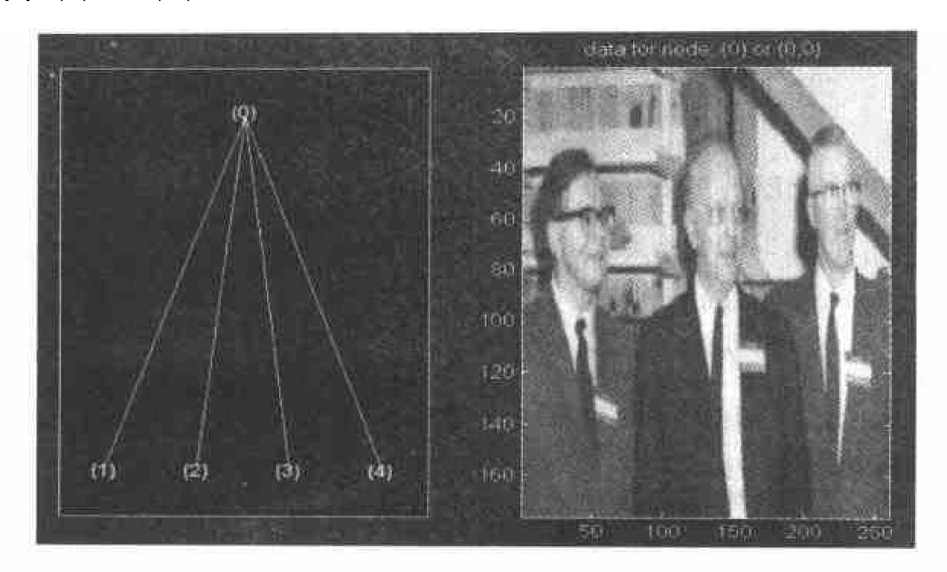


图 4.7　运行结果 1

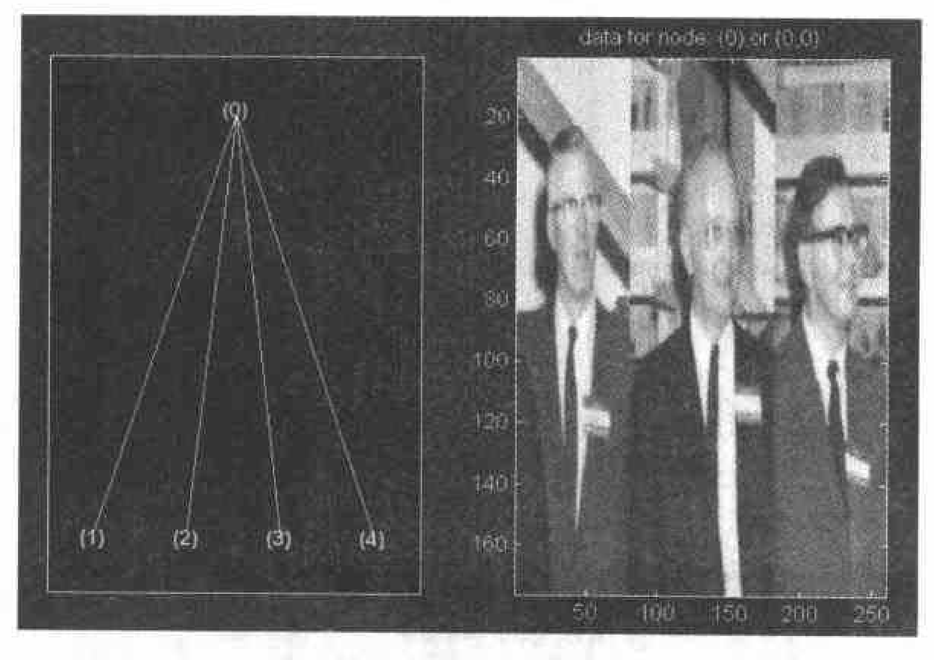


图 4-8　运行结果 2

4.2.4　小波包阈值处理（Thresholding Wavelet Packets）

例程 4-4 介绍怎样进行小波包阈值处理。

例程 4-4

```
load noisbloc
x = noisbloc;
t = wpdec(x,3,'sym4');
plot(t);
%在运行上面命令后出现的图形窗口中将 Node Label 从 Depth-postion 切换到
```

```
%Index，并单击索引值为 0 的节点，即根节点，如图 4-9 所示
%全局阈值处理
t1 = t;
sorh = 'h';
%硬阈值
thr = wthrmngr('wp1ddenoGBL','penalhi',t);
%设置消噪阈值
cfs = read(t,'data');
cfs = wthresh(cfs,sorh,thr);
t1 = write(t1,'data',cfs);
plot(t1)
%在运行上面命令后出现的图形窗口中再将 Node Label 从 Depth-postion 切换到
%Index，并单击索引值为 0 的节点，即根节点，如图 4-10 所示
%各个节点分别进行阈值处理
t2 = t;
sorh = 's';
thr(1) = wthrmngr('wp1ddenoGBL','penalhi',t);
thr(2) = wthrmngr('wp1ddenoGBL','sqtwologswn',t);
tn = leaves(t);
for k=1:length(tn)
node = tn(k);
cfs = read(t,'data',node);
numthr = rem(node,2)+1;
cfs = wthresh(cfs,sorh,thr(numthr));
t2 = write(t2,'data',node,cfs);
end
plot(t2)
%在运行上面命令后出现的图形窗口中再将 Node Label 从 Depth-postion 切换到 Index 并单击
%索引值为 0 的节点，即根节点，如图 4-11 所示
```

运行结果如图 4-9～图 4-11 所示。

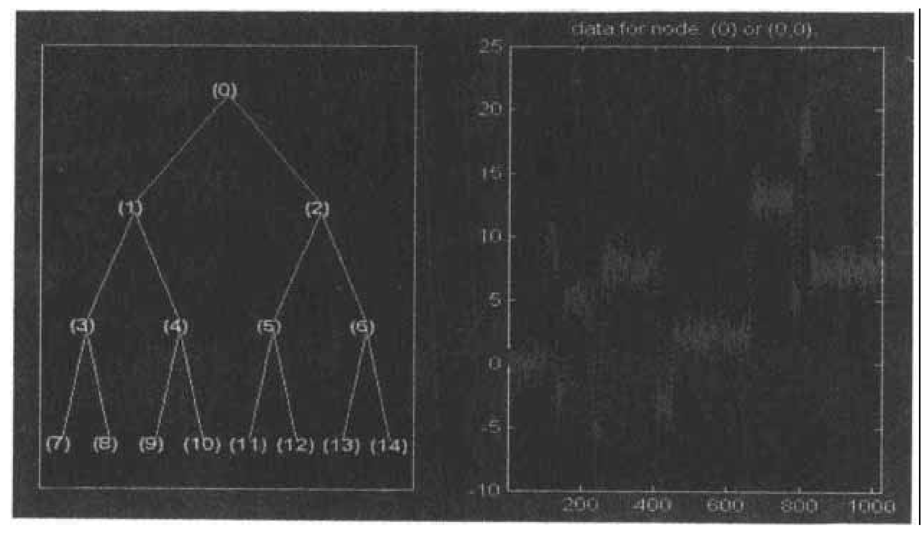


图 4-9　运行结果 1

图 4-10 运行结果 2

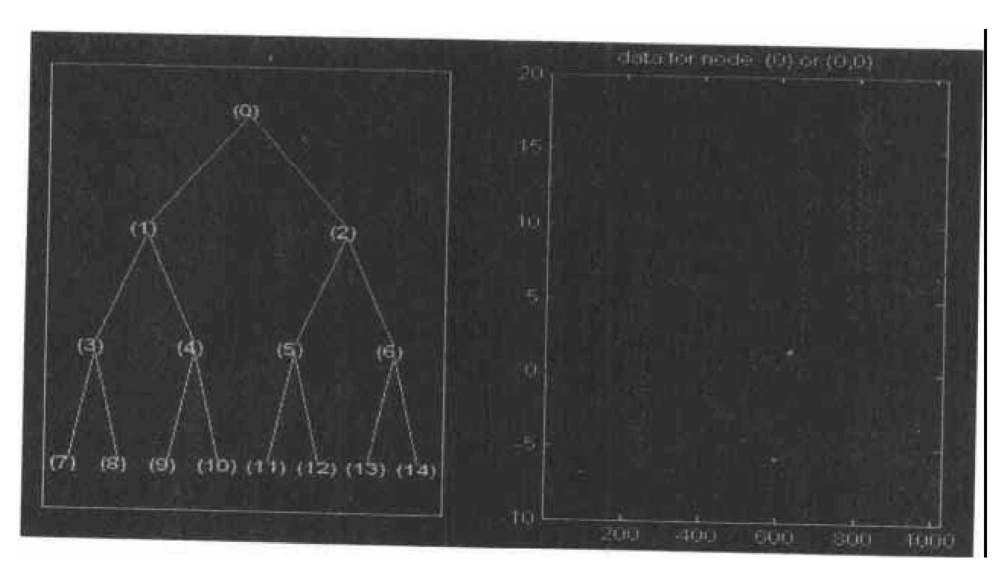


图 4-11 运行结果 3

4.3 对象的进一步讨论

下面对前面提到的小波工具箱里的 4 类对象进行进一步的讨论。

4.3.1 WTBO

类 WTBO 没有父类，其域和方法如表 4-1 和表 4-2 所示。

表 4-1 WTBO 域

wtboInfo	对象信息（Object information）
ud	用户数据（Userdata field）

表 4-2　WTBO 方法

wtbo	构造类 WTBO
get	获得 WTBO 的域内容
getwtbo	获得域内容
set	设置 WTBO 的域内容
setwtbo	设置域内容

由于小波工具箱里的所有对象都由父类 WTBO 派生而来，所以可以利用域'ud'将自己的数据同一个对象联系起来，然后去访问。

例如：假设 Obj 是由 WTBO 派生出来的对象，则：

```
Obj = set(Obj,'ud',MyData)
%定义数据
MyData = get(O,'ud')
%接受数据
```

4.3.2　NTREE

其父类是 WTBO，域、方法和私有方法如表 4-3、表 4-4 和表 4-5 所示。

表 4-3　NTREE 域

wtbo	父类
order	树的叉数
depth	树的深度
spsch	节点分解方案
tn	终节点列向量

表 4-4　NTREE 方法

ntree	构造类 ntree
findactn	查找活动节点
get	获取域内容
nodejoin	重组节点
nodesplt	分割节点
plot	画 NTREE 类
set	设置 NTREE 域内容
tlabes	树节点标志
wtreemgr	管理树结构

表 4-5　NTREE 私有方法

locnumcn	子节点的局部数
tabofasc	节点的权重表

4.3.3 DTREE

其父类是 NTREE，如表 4-6 和表 4-7 所示。

表 4-6 DTREE 域

ntree	父类
allNI	所有节点信息
terNI	终节点信息

表 4-7 DTREE 方法

dtree	构造类 DTREE
expand	扩展数据树
fmdtree	域管理
nodejoin	重组节点
nodesplt	分解节点
modcoef	重构节点系数
defaninf	定义节点信息
get	获取域信息
plot	作图
read	读取域
set	设置域内容
write	改写域
merge	重组节点数据
recons	重构节点系数
split	分解终节点数据



（1）在构造类 DTREE 的一个具体对象后，表 4-7 中在第二个方法不能被重载，第三个方法可以被重载，而第三个之后的方法必须被重载以改组、重构或分解节点数据。

（2）方法 nodejoin 调用 merge，nodesplt 调用 split，modcoef 调用 recons。

（3）为了定义节点信息，必须重载 defaninf。对每一个节点 N，基本信息为：

allNI(N,1:3): [index,size(1,1),size(1,2)]；用户可以对 allNI 添加列来增加信息。

（4）如果 get 没有重载，采用方法 getwtbo 可以获得对象的部分域内容，例如：如果 T 是由 2 阶的 DTREE 派生出的对象，Tfield 是 T 的一个域，其内容是 Tval，[a,b]= get(t,'order','Tfield')，返回 a = 2，b = 'errorWTBX'。显然这是错误的信息。采用[a,b] = getwtbo(t,'order','Tfield')，则返回正确的结果：a = 2，b = Tval。

4.3.4 WPTREE

其父类是 DTREE，如表 4-8 和表 4-9 所示。

表 4-8 WPTREE 域

dtree	父类
wavInfo	小波信息
entInfo	熵信息

说明

wavInfo 结构如下：

wavName 小波基函数名
Lo_D 低通分解滤波器
Hi_D 高通分解滤波器
Lo_R 低通重构滤波器
Hi_R 高通重构滤波器

entInfo 结构如下：

entName 熵名称
entPar 熵参数

表 4-9 WPTREE 方法

wptree	构造类 wptree
下面的方法是从父类 DTREE 继承而来	
defaninf	功能请参见前面说明
get	
merge	
read	
recons	
set	
split	
tlabels	
write	
下面是 WPTREE 私有方法	
bestlevt	计算最佳完整小波包树
besttree	计算最佳树
entrupd	熵更新
wp2wtree	从小波包树中提取小波树
wpcoef	小波包系数
wpcutree	剪切小波包分解树

（续表）

wpjoin	重新组合小波包
wpplotcf	小波包染色系数作图
wprcoef	重构小波包系数
wprec	一维小波包重构
wprec2	二维小波包重构
wpsplt	分解小波包
wpthcoef	小波包系数阈值处理
wpviewcf	小波包染色系数作图

4.4　对象的高级应用

下面是几个利用新的对象来扩展小波工具箱功能的例子。

【例 4-1】构造一个小波树（WTREE）对象

这个例子通过重载类 DTREE 的 split 和 merge 方法，定义了一个新的类 WTREE，如表 4-10 和表 4-11 所示。

表 4-10　WTREE 域

dtree	父类
dwtMode	DWT 延拓模式
waveInfo	小波信息

表 4-11　WTREE 方法

wtree	构造类 WTREE
merge	合并节点数据
split	分解节点数据

读者可以通过 M 文件编辑器打开位于 MATLAB 6.5 安装目录下的 toolbox/wavelet/wavedemo 子目录中的 ex1_wt（一维）和 ex2_wt（二维）两个例子来学习如何建立新的面向对象特征的函数。

【例 4-2】建立新的右小波树类（RWVTREE）

我们从【例 4-1】定义的新类 WTREE 出发，通过重载其方法 split、merge 和 plot（从 DTREE 继承而来），定义如表 4-12 和 4-13 所示。

表 4-12　RWVTREE 域

dummy	不用
wtree	父类

表 4-13　RWVTREE 方法

rwvtree	构造类 RWVTREE
merge	合并节点数据
plot	类 RWVTREE 作图
split	分解节点数据

用户可以在命令行窗口运行命令 ex1_rwvt，并单击节点 14，结果如图 4-12 所示（彩色效果图见彩插 7）。可以看出，低频信号以黄色显示，高频信号以红色显示。

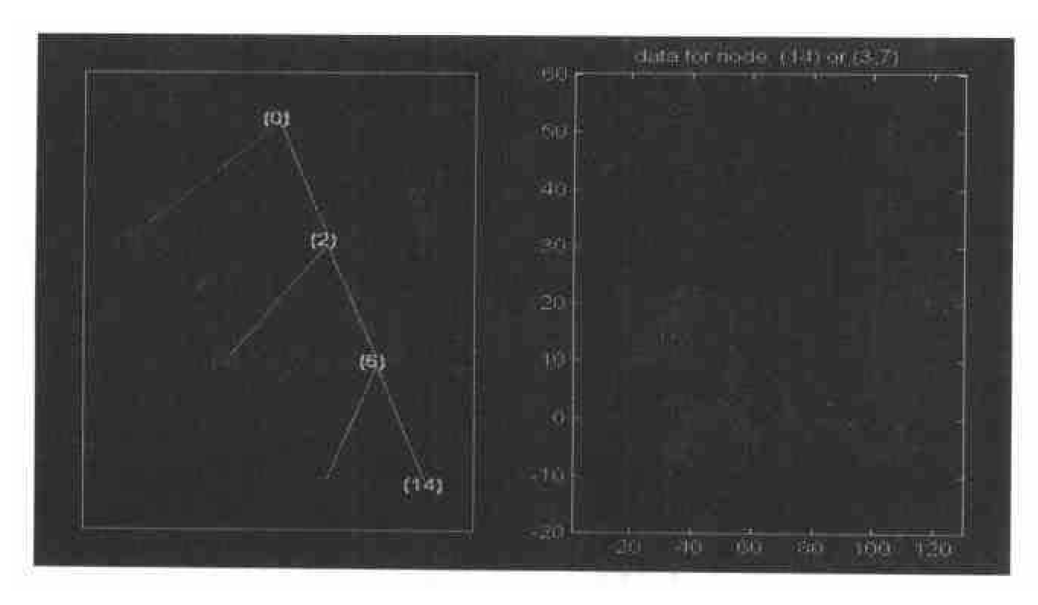


图 4-12　运行结果

【例 4-3】 建立新的小波树类（WVTREE）

我们仍从【例 4-1】定义的新类 WTREE 出发，通过重载其方法 get、plot 和 recons（从 DTREE 继承而来），以及来自类 WTREE 本身的 split 和 merge 方法，定义如表 4-14 和表 4-15 所示。

表 4-14　WVTREE 域

dummy	不用
wtree	父类

表 4-15　WVTREE 方法

wvtree	构造类 WVTREE
get	获取 WVTREE 的域内容
plot	类 RWVTREE 作图
recons	重构节点系数

用户可以在命令行窗口运行命令 ex2_wvt，并单击节点 2，结果如图 4-13 所示（彩色效果图见彩插 8）。

可以看出，图形低频部分以黄色显示，高频部分以红色显示。图形的标题包含了 DWT 延拓模式，即'sym'（对称延拓）。

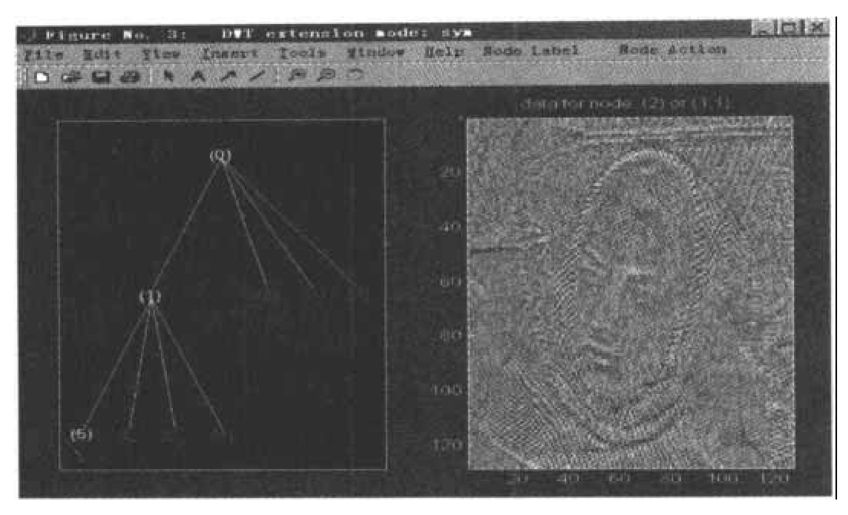


图 4-13　运行结果

第 5 章　小波函数的添加

在 MATLAB 6.5 小波工具箱里系统提供了 15 个小波函数，用户可以通过命令 waveinfo 获取关于这些已有小波函数的特性介绍。但在实际应用中，有时这些小波并不能满足用户特定的分析需求，必须使用用户自己定义的小波函数，这就面临一个如何将用户自己定义的小波函数添加到小波工具箱里的问题。用户可以利用系统通过的小波管理函数 wavemngr 来完成这一功能。同前面第 2 章一样，系统允许用户采用命令窗口方式和图形工具方式这两种方式来定义新的小波函数。

值得注意的是，在添加用户自己的小波函数之前，用户必须仔细检查所要添加的小波函数是否满足作为一个小波函数所必须具有的各种数学性质，因为小波工具箱并不做这一检验。这一点必须很小心。

本章主要内容：

- 小波函数的添加
- 小波函数系列的添加
- 添加小波函数系列之后的工作

5.1　小波函数的添加

在采用命令 wavemngr 添加一个小波函数之前，必须按照以下 6 个步骤做好准备工作，下面我们分步逐一进行介绍。

5.1.1　选择小波函数的全名称（full name）

小波函数的全名称（简称fn）必须是一个字符串，如系统已有的15个小波函数：Haar，Daubechies，Symlets，Coiflets，BiorSplines，ReverseBior，Meyer，Dmeyer，Gaussian，Mexican_hat，Morlet，，Complex Gaussian，Shannon，Frequency B-Spline和Complex Morlet。

5.1.2　选择小波函数的缩写名（short name）

小波函数的缩写名（简称fsn）必须是一个字符个数小于或等于4的字符串，如与上面列出的小波函数全名称对应的缩写名依次为：haar，db，sym，coif，bior，rbio，meyr，dmey，gaus，mexh，morl，cgau，shan，fbsp和cmor。

5.1.3　选择小波类型

通常有如下 5 类小波类型可选择：

- 具有有限冲激响应滤波器（FIR）的正交小波

这种小波函数可以通过尺度滤波器 w 进行定义，在 MATLAB 6.5 中已有的该类小波函数有：Haar，Daubechies，Coiflets 和 Symlets。

- 具有有限冲激响应滤波器（FIR）的双正交小波

这种小波函数可以通过用于小波重构的尺度滤波器 wr 和用于小波分解的尺度滤波器 wr 进行定义，在 MATLAB 6.5 中已有的该类小波函数有：BiorSplines。

- 不具有有限冲激响应滤波器（FIR）但具有尺度函数的正交小波

这种小波函数可以通过小波函数和尺度函数的定义来进行定义，在 MATLAB 6.5 中已有的该类小波函数有：Meyer。

- 不具有有限冲激响应滤波器（FIR）也不具有尺度函数的小波

这种小波可以通过小波函数的定义来进行定义，在 MATLAB 6.5 中已有的该类小波函数有：Morlet 和 Mexican_hat。

- 有限冲激响应滤波器（FIR）和尺度函数都不具有的复小波

这种小波可以通过小波函数的定义来进行定义，在 MATLAB 6.5 中已有的该类小波函数有：Complex Gaussian 和 Shannon。

5.1.4 定义在给定小波家族里的序列号

如果在一个小波函数系中包含很多小波，则该小波函数系中的小波函数名由小波函数缩写名和一个序列号构成加以区分。字符串参数 nums 包含一个小波函数序列的所有序列号，各个序列号之间用空格隔开。对于只有单个小波函数的小波函数系（如 Haar、Meyer 和 Morlet），该参数并不使用。

例如，对于 BiorSplines 小波函数系列：

```
fsn = 'bior'
nums = '1.1 1.3 1.5 2.2'
```

则可以产生 4 个小波函数 bior1.1、bior1.3、bior1.5 和 bior2.2。

5.1.5 建立相应的*.mat 文件或*.m 文件

函数 wavemngr 需要一个包含*.mat 文件名或*.m 文件名的字符串参数。如果一个小波函数系包含许多个小波函数，则必须定义一个*.m 文件，并且对于不同的小波函数类型，该文件也必须具有不同的类型。如果一个小波函数系只含一单个小波函数，则对于上面第一类型的小波函数必须定义一个*.mat 文件。该文件必须以小波函数的缩写名作为文件名，而且必须包含一个名称为 fsn、数值为尺度滤波器的变量参数。

下面是与上面提到的 5 类小波相应的*.m 文件详细定义。

1. 第一类小波（具有有限冲激响应滤波器（FIR）的正交小波）

*.m 文件第一行的语法结构是：

```
function w=file(wname)
```

这里输入参数 wname 是一个包含小波函数名的字符串，输出参数 w 是相应的尺度滤

波器，而且 w 的长度必须为偶数，否则小波工具箱自动对其补 0 以满足这一要求。

小波工具箱中系统提供的小波函数系中，尺度滤波器的数值总和为 1。对于一个新的小波函数（其尺度滤波器的数值不能为 0），可以不进行正规化，要求小波工具箱会自动对尺度滤波器进行正规化处理。

MATLAB 中已定义的这种类型的*.m 文件例子有：

- Daubechies 小波函数系的 dbwavf.m 文件；
- Coiflets 小波函数系的 coifwavf.m 文件；
- Symwavf 小波函数系的 symwavf.m 文件。

2．第二类小波（具有有限冲激响应滤波器（FIR）的双正交小波）

*.m 文件第一行的语法结构是：

function [wr,wd]=file(wname)

这里输入参数 wname 是一个包含小波函数名的字符串，输出参数 wr 和 wd 分别是相应的重构和分解尺度滤波器。wr 和 wd 必须具有相同的偶数尺度，通常初始的双正交滤波器不满足该条件，小波工具箱会自动补 0 以满足这一要求。

同上，小波工具箱中系统提供的小波函数系中，尺度滤波器的数值总和为 1。对于一个新的小波函数（其尺度滤波器的数值不能为 0），可以由小波工具箱自动对其进行正规化处理。

MATLAB 中已定义的这种类型的*.m 文件例子有：

BiorSplines 函数系的 biorwavf.m 文件。

3．第三类小波（具有尺度函数但不具有有限冲激响应滤波器（FIR）的正交小波）

*.m 文件第一行的语法结构是：

function [phi,psi,t]=file(lb,ub,n,wname)

这里输出参数 phi 和 psi 分别是相应的支撑长度为[lb,ub]的尺度函数和小波函数。参数 wname 是可选的。

MATLAB 中已定义的这种类型的*.m 文件例子有：Meyer 函数系的 meyer.m 文件。

4．第四、五类小波（尺度函数和有限冲激响应滤波器（FIR）都不具备的正交小波）

*.m 文件第一行的语法结构是：

function [psi,t]=file(lb,ub,n,wname) 或

function [psi,t]=file(lb,ub,n,wname,“可选参数”)

这里输出参数 psi 是相应的支撑长度为[lb,ub]的小波函数。wname 是可选的。

MATLAB 中已定义的第四种类型的*.m 文件例子有：

- Mexican_hat 函数系的 mexihat.m 文件；
- Morlet 函数系的 morlet.m 文件。

MATLAB 中已定义的第五种类型的*.m 文件例子有：

- Shannon 函数系的 shanwavf.m；
- Complex Morlet 函数系的 cmorwavf.m 文件。



对第三、四和五类型的小波，参数 wname 是可选的。只有要添加的新的小波系含有一系列小波和在图形工具接口（GUI）中要用到该小波时才必须使用该参数。对第四、五类型的小波，用户可以通过 MATLAB 的帮助功能查看函数 fbspwavf 来找到一个使用“可选参数”的例子。

5.1.6 定义有效支撑长度

对于第三、四和五类型的小波函数，由于它们不具有紧支撑性，所以必须定义其有效支撑长度。其含义就是指明该小波函数的上下界，例如对于系统已有的小波函数，其有效支撑长度如表 5-1 所示。

表 5-1 小波函数的有效支撑长度

小波函数族（Family）	下界（Low Bound）	上界（Upper Bound）
Meyer	−8	8
Mexican_hat	−5	5
Morlet	−4	4

5.2 小波函数系列的添加

要添加一个新的小波函数系列，可以通过下面 wavemngr 函数的两种形式来进行：

- wavemngr('add',fn,fsn,wt,nums,file)
- wavemngr('add',fn,fsn,wt,nums,file,b)

例如：

（1）wavemngr('add','Ndaubechies','ndb',1,'1 2 3 4 5','dbwavf');

（2）wavemngr('add','Ndaubechies','ndb',1,'1 2 3 4 5 * *','dbwavf');

（3）wavemngr('add','Nbiorwavf','nbio',2,'1.1 1.3','biorwavf');

（4）wavemngr('add','Nmeyer','nmey',3,'','meyer',[-8,8]);

（5）wavemngr('add','Nmorlet','nmor',4,'','morlet',[-4,4])

下面是两个采用函数wavemngr添加新的小波系列的具体例子。

【例 5-1】 向系统添加由 Strang 和 Nguyen 提出的一个小波函数系列。

小波函数系列全名称：Binlets

小波函数系列的缩写名：binl

小波类型：第二类，即具有FIR滤波器的双正交小波

小波函数系列中的序列号有：7.9（这儿我们只举一个说明）

下面是用来产生该小波函数滤波器的*.m文件名为binlwavf.m，用户可以在MATLAB 6.5安装目录下的toolbox\wavelet\wavedemo子目录中找到，我们对其说明如下：

```
function [Rf,Df] = binlwavf(wname)
%BINLWAVF Biorthogonal wavelet filters (binary wavelets: Binlets).
%    [RF,DF] = BINLWAVF(W) 返回两个尺度滤波器，其中
%    RF 是重建滤波器
%    DF 是分解滤波器
%    双正交小波函数由字符串 W 指明
%    W = 'binlNr.Nd' 其中 Nr 和 Nd 的可取值为
%    Nr = 7   Nd = 9
%    M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 12-Mar-96.
%    Last Revision: 17-Apr-1998.
%    Copyright 1995-2000 The MathWorks, Inc.
% $Revision: 1.9 $
% 参数校验
if errargn(mfilename,nargin,[0 1],nargout,[0:2]), error('*'); end
%小波函数的扩展语句
Nr = 7; Nd = 9;
% for possible extension
% more wavelets in 'Binlets' family
%----------------------------------
if nargin==0
    Nr = 7; Nd = 9;
elseif   isempty(wname)
    Nr = 7; Nd = 9;
else
    if isstr(wname)
        lw = length(wname);
        ab = abs(wname);
        ind = find(ab==46 | 47<ab | ab<58);
        li = length(ind);
        err = 0;
        if li==0
            err = 1;
        elseif ind(1)~=ind(li)-li+1
            err = 1;
        end
        if err==0 ,
            wname = wstr2num(wname(ind));
            if isempty(wname) , err = 1; end
        end
    end
    if err==0
        Nr = fix(wname); Nd = 10*(wname-Nr);
    else
        Nr = 0; Nd = 0;
    end
end
```

```
% 函数扩展语句和出错测试
%-------------------------------------------
if Nr~=7 , Nr = 7; end
if Nd~=9 , Nd = 9; end

if Nr == 7
    if Nd == 9
        Rf = [-1 0 9 16 9 0 -1]/32;
        Df = [ 1 0 -8 16 46 16 -8 0 1]/64;
    end
end
```

以下这段程序旨在添加该小波函数系列：

```
% 添加一个新的双正交小波函数系列
wavemngr('add','Binlets','binl',2,'7.9','binlwavf')
% 列出所有的小波函数系列，查看是否添加成功
wavemngr('read')
ans =
===================================================================
Haar haar
Daubechies db
Symlets sym
Coiflets coif
BiorSplines bior
ReverseBior rbio
Meyer meyr
DMeyer dmey
Gaussian gaus
Mexican_hat mexh
Morlet morl
Complex Gaussian cgau
Shannon shan
Frequency B-Spline fbsp
Complex Morlet cmor
Binlets binl
```

如果要获得该小波函数系列的在线帮助信息，可以建立如下的帮助文件：

```
function binlinfo
%BINLINFO 双正交小波函数系列的信息(binlets).
% Family Binlets
% Short name binl
% Order Nr,Nd Nr = 7 , Nd = 9
%
% Orthogonal no
```

```
% Biorthogonal yes
% Compact support yes
% DWT possible
% CWT possible
%
% binl    Nr.Nd          ld                  lr
%                  effective length     effective length
%                    of LoF_D              of HiF_D
% binl     7.9            7                    9
```

【例 5-2】 加入一个新的紧支撑的双正交小波函数系列。它是 Daubechies 小波函数系列的一个派生小波函数系列，其产生基于 Bernstein 多项式。

小波函数系列全名称：Lemarie

小波函数系列的缩写名：lem

小波类型：第一类，即具有FIR滤波器的正交小波

小波函数系列中的序列号有：1、2、3、4、5

下面是用来产生该小波函数滤波器的*.m 文件名为 lemwavf.m，用户可以在 MATLAB 6.5 安装目录下的 toolbox\wavelet\wavedemo 子目录中找到：

```
function F = lemwavf(wname)
%LEMWAVF Lemarie wavelet filters.
%    F = LEMWAVF(W) returns the scaling filter
%    associated with Lemarie wavelet specified
%    by the string W, where W = 'lemN'.
%    Possible values for N are:
%            N = 1, 2, 3, 4 or 5.
%    N.B:
%        Using the MATLAB Extended Symbolic Toolbox
%        possible values of N are positive integers.

%    M. Misiti, Y. Misiti, G. Oppenheim, J.M. Poggi 12-Mar-96.
%    Last Revision: 17-Apr-1998.
%    Copyright 1995-2000 The MathWorks, Inc.
%    $Revision: 1.10 $ $Date: 2000/06/08 13:43:35 $

if isstr(wname) & ~isempty(wname)
        lw = length(wname); ab = abs(wname);
        ii = lw+1;
        while (47<ab(ii-1)) & (ab(ii-1)<58) , ii = ii-1; end
        num = wstr2num(wname(ii:lw));
else
        num = wname;
end
if isempty(num) | any(num < 1) | any(num ~= fix(num))
        error('*** Invalid wavelet number ! ***');
```

```
end

switch num
    case 1
F = [...
    0.46069299844871   0.53391629051346   0.03930700681965   -0.03391629578182 ...
];

    case 2
F = [...
    0.31555164655258   0.59149765057882   0.20045477817080   -0.10034811856888 ...
   -0.01528128420694   0.00846362066021   -0.00072514051618   0.00038684732960 ...
        ];

    case 3
F = [...
    0.23108942231941   0.56838231367966   0.33173980738190   -0.09447000132310 ...
   -0.06203683305244   0.02661631105889   -0.00209952890579   0.00001769381066 ...
    0.00128429679795   -0.00053703458679   0.00002283826072 -0.00000928544107 ...
        ];

    case 4
F = [...
    0.17565337503255   0.52257484913870   0.42429244721660   -0.04601056550580 ...
   -0.11292720306517   0.03198741803409   0.00813124691980   -0.00743764392677 ...
    0.00548090619143 -0.00140066128481   -0.00054200083128   0.00025607264164 ...
   -0.00008795126642   0.00003025515674   -0.00000082014466   0.00000027569334 ...
        ];

    case 5
F = [...
    0.13807658847623   0.47310642622099   0.48217097800239   0.02112933622031 ...
   -0.15081998732499   0.01935767268926   0.02716532750995   -0.01588522540421 ...
    0.00671209165995   0.00120022744496   -0.00321203819186   0.00115266788547 ...
   -0.00018266213413   -0.00002953360842   0.00008433396295   -0.00002997969339 ...
    0.00000534552866   -0.00000159098026   0.00000003069431 -0.00000000895816 ...
        ];

    otherwise
        % compute bernstein polynomial of order 4*num-1.
        % requires the Extended Symbolic Toolbox.
        if ~exist('maple')
                msg = '*** The Extended Symbolic Toolbox is required ***';
                error(msg)
                break
        end
```

```
            order = 4*num-1;
            mpa('ord',order);
            maple('readlib(bernstein):');
            maple('f:=proc(t) if t<1/4 then 0 else if t >3/4 then 1 else 2*t-1/2 fi fi end:');
            cfs = maple('bernstein(ord,f,(1+x)/2);');
            ber = sym2poly(cfs);

            r = roots(ber);
            v = r-sqrt(r.^2-1);
            ind = find(abs(v)>1);
            if ~isempty(ind)
                v(ind)=ones(size(ind))./v(ind);
            end
            F = real(poly(v));
            F = F/sum(F);
end
```

下面是添加该小波函数系列的程序清单：

```
%将系统已有的小波函数系列全部列出
wavemngr('read')
ans =
Haar                    haar
Daubechies              db
Symlets                 sym
Coiflets                coif
BiorSplines             bior
ReverseBior             rbio
Meyer                   meyr
DMeyer                  dmey
Gaussian                gaus
Mexican_hat             mexh
Morlet                  morl
Complex Gaussian        cgau
Shannon                 shan
Frequency B-Spline      fbsp
Complex Morlet          cmor
%列出上面各小波函数系列中的小波函数
wavemngr('read',1)
ans =
==================================
Haar                    haar
==================================
Daubechies              db
-----------------------------
db1   db2   db3   db4
```

```
db5   db6   db7   db8
db9   db10  db**
===================================
Symlets                   sym
------------------------------
sym2 sym3 sym4 sym5
sym6 sym7 sym8 sym**
===================================
Coiflets                  coif
------------------------------
coif1 coif2 coif3 coif4
coif5
===================================
BiorSplines               bior
------------------------------
bior1.1     bior1.3     bior1.5     bior2.2
bior2.4     bior2.6     bior2.8     bior3.1
bior3.3     bior3.5     bior3.7     bior3.9
bior4.4     bior5.5     bior6.8
===================================
ReverseBior               rbio
------------------------------
rbio1.1     rbio1.3     rbio1.5     rbio2.2
rbio2.4     rbio2.6     rbio2.8     rbio3.1
rbio3.3     rbio3.5     rbio3.7     rbio3.9
rbio4.4     rbio5.5     rbio6.8
===================================
Meyer                     meyr
===================================
DMeyer                    dmey
===================================
Gaussian                  gaus
------------------------------
gaus1gaus2gaus3gaus4
gaus5gaus6gaus7gaus8
gaus**
===================================
Mexican_hat               mexh
===================================
Morlet                    morl
===================================
Complex Gaussian          cgau
------------------------------
cgau1       cgau2       cgau3       cgau4
cgau5       cgau**
===================================
```

```
Shannon                   shan
------------------------------
shan1-1.5  shan1-1    shan1-0.5  shan1-0.1
shan2-3    shan**
==================================
Frequency B-Spline        fbsp
------------------------------
fbsp1-1-1.5fbsp1-1-1  fbsp1-1-0.5fbsp2-1-1
fbsp2-1-0.5fbsp2-1-0.1fbsp**
==================================
Complex Morlet            cmor
------------------------------
cmor1-1.5  cmor1-1    cmor1-0.5  cmor1-1
cmor1-0.5  cmor1-0.1  cmor**
%加入新的正交小波函数系列
wavemngr('add','Lemarie','lem',1,'1 2 3 4 5','lemwavf');
% 二进制文件'wavelets.asc'被另存为文件'wavelets.prv'，并且被修改产生 mat 文件
%'wavelets.inf'
%列出所有的小波函数系列，查看是否添加成功
wavemngr('read')
ans =
==================================
Haar                      haar
Daubechies                db
Symlets                   sym
Coiflets                  coif
BiorSplines               bior
ReverseBior               rbio
Meyer                     meyr
DMeyer                    dmey
Gaussian                  gaus
Mexican_hat               mexh
Morlet                    morl
Complex Gaussian          cgau
Shannon                   shan
Frequency B-Spline        fbsp
Complex Morlet            cmor
Lemarie                   lem
```

可以看出，小波函数系列 lem 已经被成功添加到系统中。

5.3 添加小波函数系列之后的工作

采用 wavemngr 函数添加一个小波函数系列之后，工具箱自动在当前目录下生成三个文件：两个 ASCII 码文件 wavelets.asc 和 wavelets.prv，以及一个*.mat 类型的文件

wavelets.inf。

如果希望在小波工具箱里使用自己定义的小波函数系列，可以按下述步骤进行：

（1）创建一个新的目录来存放自己的小波函数系列；

（2）把上面的三个相关文件移动到该目录；

（3）利用命令 path 将这个新建的目录放入 MATLAB 6.5 的搜索路径中；

（4）最好在同一个目录中修改自己定义的小波函数系列，每次都将生成的小波函数系列放在不同的目录中可能会导致不可预料的结果；

（4）定义一个名为<fsn>info.m 的*.m 文件，例如系统已有的 dbinfo.m 或 morlinfo.m 文件。

完成以上步骤后，在小波工具箱中的图形接口界面中单击【Wavelet Family】按钮时，这个文件就会自动出现在小波显示选择框中。

第 6 章　小波分析用于信号处理

前面几章我们介绍了小波分析的基础理论和如何利用 MATLAB 6.5 语言进行基本的小波和小波包分析，但还没有涉及到工程实际。下面两章我们将以工程实际中可能遇到的一些信号处理需求为背景，来介绍小波分析在工程实际信号处理中的应用方法。

本章主要内容：

- 常用信号的小波分析
- 信号的特征提取
- 信号处理
- 应用图形界面（GUI）方式进行信号处理

6.1 概　述

众所周知，在信号处理领域中有着许多的方法和工具，其中最为常用的就是傅里叶变换（Fourier Transform，FT）。虽然 FT 能够将信号的时域特征和频域特征联系起来，且能分别从信号的时域和频域中对其进行观察，但是它却不能将二者有机地结合起来。这主要是因为在信号的时域波形中不包含其频域信息，而信号的傅里叶谱是信号的统计特性，它是整个时间域内的积分，它不具有局部化分析信息的功能。后来所发展的短时傅里叶变换（Short Time Fourier Transform，STFT），其基本思想是：把信号划分成许多小的时间间隔，再用 FT 分析每一个时间间隔，以便确定该时间间隔内所包含的信号频率。STFT 用来分析平稳信号时，其效果尚可。对于非平稳信号，在波形变化比较平缓时，其主频是低频，此时则要求有较高的频率分辨率；而在波形变化比较剧烈时，其主频是高频，此时则要求有较高的时间分辨率。但是 STFT 是一个窗口固定的分析方法。因此在这种非平稳信号的情况下，STFT 不可能兼顾上述两方面的要求，暴露出它的不足。

小波变换（Wavelet Transform）是一种窗口大小不变但形状可变，即时间窗和频率窗都可改变的时频局部化分析方法。即在高频部分具有较高的时间分辨率和较低的频率分辨率，在低频部分具有较低的时间分辨率和较高的频率分辨率，因此小波分析被誉为数学显微镜，正是这种特性，使它具有对信号的自适应性，因而越来越广泛地被应用于工程实际。由于小波变换本质上是一个范围可变的窗口方法，它可以用较长的时间间隔来获取更精确的低频信息，用较短的时间间隔来获取高频信息。又由于小波分析具有局部分析和细化的功能，所以小波分析可以揭示其他信号分析方法所丢失的数据信息，如断点、高阶导数不连续性、趋势和自相似性等。而且与传统的信号分析技术相比，小波分析还能在没有明显损失的情况下，对信号进行消噪和压缩。

6.2 常用信号的小波分析

在这一节，我们将给出一些常见的信号形式，并对其进行小波分析，总结出小波分析分别对这些信号有效的处理范围。

在正式讲解之前，先来看看这些信号的数学表示式：

（1）正弦波的线性组合：

$$s(t) = \sin(3t) + \sin(0.3t) + \sin(0.03t)$$

（2）分段信号：

$$s(t) = \begin{cases} \sin(0.03t) & 1 \leqslant t \leqslant 500 \\ \sin(0.3t) & 501 \leqslant t \leqslant 1000 \end{cases}$$

（3）对称区间上的均匀白噪声：

$$b_1(t) \qquad t \in [-0.5, 0.5]$$

（4）有色、三阶自适应模型噪声：

$$b_2(t) = -1.5b_2(t-1) - 0.75b_2(t-2) - 0.125b_2(t-3) + b_1(t) + 0.5$$

（5）染噪的多项式信号：

$$s(t) = t^2 - t + 1 + b_1(t)$$

（6）阶跃信号：

$$s(t) = \begin{cases} 0 & 1 \leqslant t \leqslant 500 \\ 20 & 501 \leqslant t \leqslant 1000 \end{cases}$$

（7）染噪的斜坡信号：

$$s(t) = \begin{cases} \dfrac{3t}{500} + b(t) & 1 \leqslant t \leqslant 500 \\ 3 + b(t) & 500 \leqslant t \leqslant 1000 \end{cases}$$

（8）染噪的正弦信号：

$$s(t) = \sin(0.03t) + b(t)$$

（9）三角波与正弦波的线性组合：

$$s(t)=\begin{cases}\dfrac{t-1}{500}+\sin(0.3t) & 1\leqslant t\leqslant 500\\ \dfrac{1000-t}{500}+\sin(0.3t) & 501\leqslant t\leqslant 1000\end{cases}$$

（10）染噪的三角波与正弦波的组合：

$$s(t)=\begin{cases}\dfrac{t-1}{500}+\sin(0.3t)+b(t) & 1\leqslant t\leqslant 500\\ \dfrac{1000-t}{500}+\sin(0.3t)+b(t) & 501\leqslant t\leqslant 1000\end{cases}$$

下面我们应用 Daubechies 小波对上述各信号进行分析。在此，给出了各个信号，以及对它们进行小波分解后所得到的近似和细节。

6.2.1　正弦波的线性组合

这个信号是由周期大约分别为 200、20 和 2 且幅值为 1 的三种正弦波叠加而成，其采样周期为 1。在此，应用 db3 小波对该信号进行 5 层分解，所得到的结果如图 6-1 所示。可以从图 6-1 中看出，细节 d1 显示了周期最小的正弦波，细节 d4 显示了周期为 20 的正弦波，而周期为 200 的正弦波则出现在近似 a4 中。之所以有这样的结果，是因为利用小波对信号进行分解时，它将信号分解为低频部分（近似，approximation）和高频部分（细节，detail）。在图中，也表示出了小波分解可得到的其他结果，如近似 a3 和 a4 之间出现了不连续。

利用小波分析，可以对类似的叠加信号进行以下几个方面的研究：

- 间断点检测
- 波形未来发展趋势的检测
- 各分信号的频率识别
- 信号从近似到细节的迁移

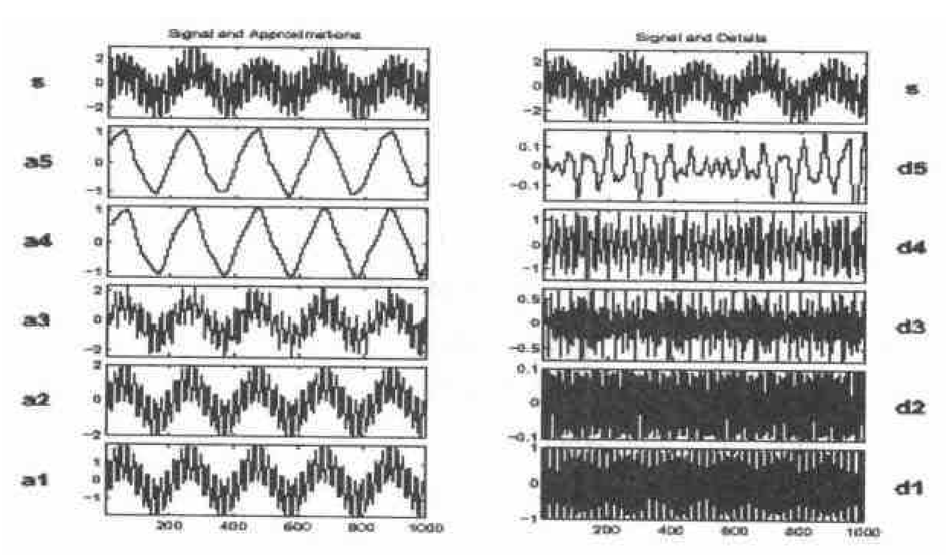


图 6-1　正弦波线性组合信号的小波分析

6.2.2 分段信号

该信号是由周期大约为 2 的“较快”正弦波和周期大约为 20 的“较慢”正弦波以连续的方式连接而成，如图 6-2 所示。

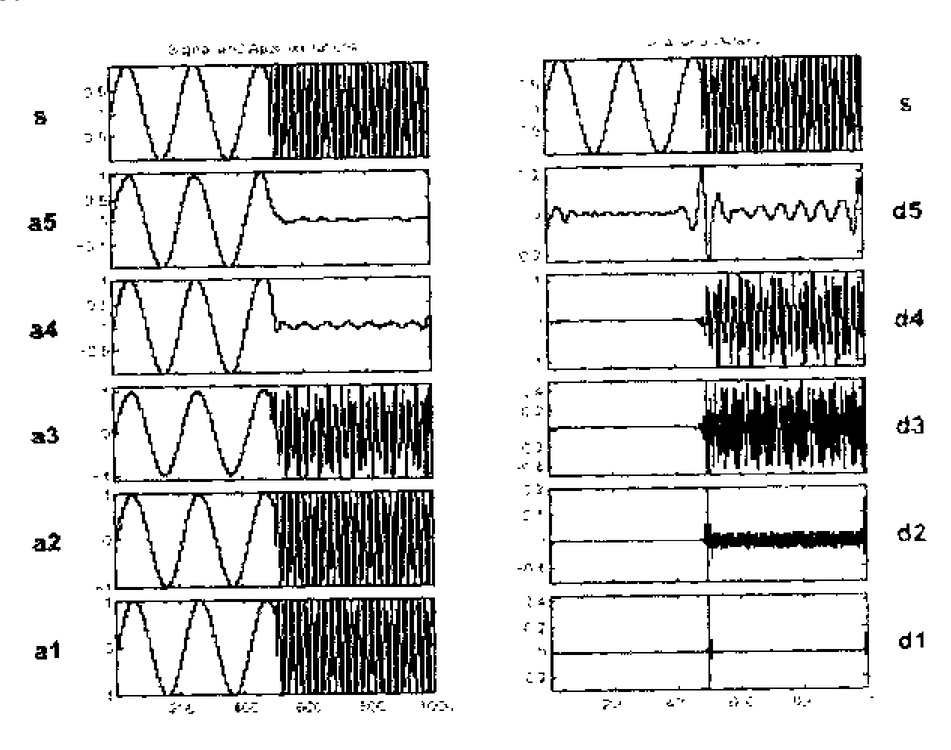


图 6-2 分段信号的小波分析

在此，利用 db5 小波对该信号进行 5 层分解。很明显，在细节 d1 和 d2 中清晰地显示出了该信号的频率间断，但是如果应用 FT 检测的话，不能够辨识出这个信号频率的瞬变。在近似 a4 和 a5 中，“较慢”的正弦波被分离出来。

小波分析对该信号的有效处理有以下两方面：

- 信号抑制
- 信号未来发展趋势的检测

6.2.3 均匀白噪声

所有噪声信号，一般都是不规则的。对于该类信号的处理，传统的方法较为繁琐。这里应用 db3 小波对均匀白噪声信号进行 5 层分解，其结果如图 6-3 所示。在图中的近似和细节的各层上，都能发现噪声信号，且表现出的信息也是不规则的。我们仔细观察图中左列的各层近似和右列的各层细节，将会发现噪声信号的不一致性在两个相邻层之间按照倍数关系递减，即 variance（D_j）=variance（D_{j-1}）/2。还应该清楚，近似和细节信号不是白噪声，而且这些信号之间的相依性会随着分解的递减而增加。另一方面，小波系数是随机的、不相关的变量。

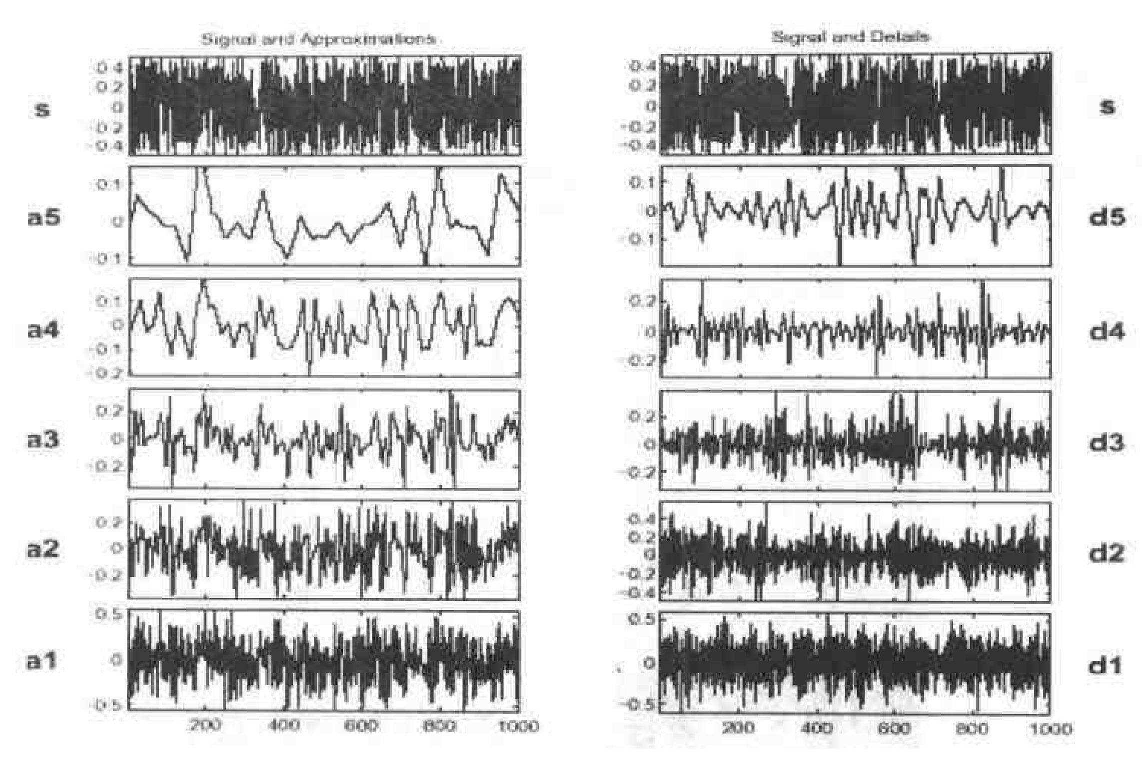


图 6-3　均匀白噪声的小波分析

图 6-4 中给出了有色噪声的小波分析结果。由于非白噪声的频谱主要集中在较高的频率部分。因此，这种噪声信号一般可在小波分解的细节 d1 中分辨。

小波分析在处理噪声信号时的有效范围为：

- 噪声处理
- 相关性与分歧的分析
- 细节中所包含频率与近似中所包含频率的比较
- 不同细节在信号处理中的相对重要性
- 在连续性分析下，辨识有色噪声的紊乱

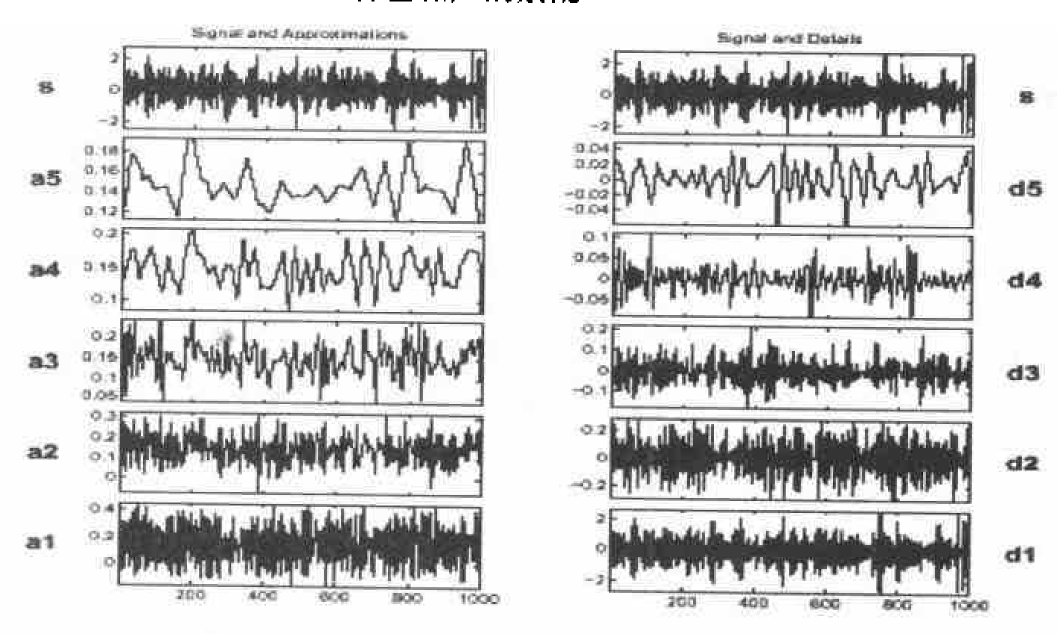


图 6-4　有色噪声的小波分析

6.2.4 染噪的多项式信号

在图 6-5 中的信号 s 是由一个二阶多项式和一个白噪声叠加而成的。分别应用 db2 和 db3 小波对该信号进行 4 层分解，所得结果如图 6-5 所示。

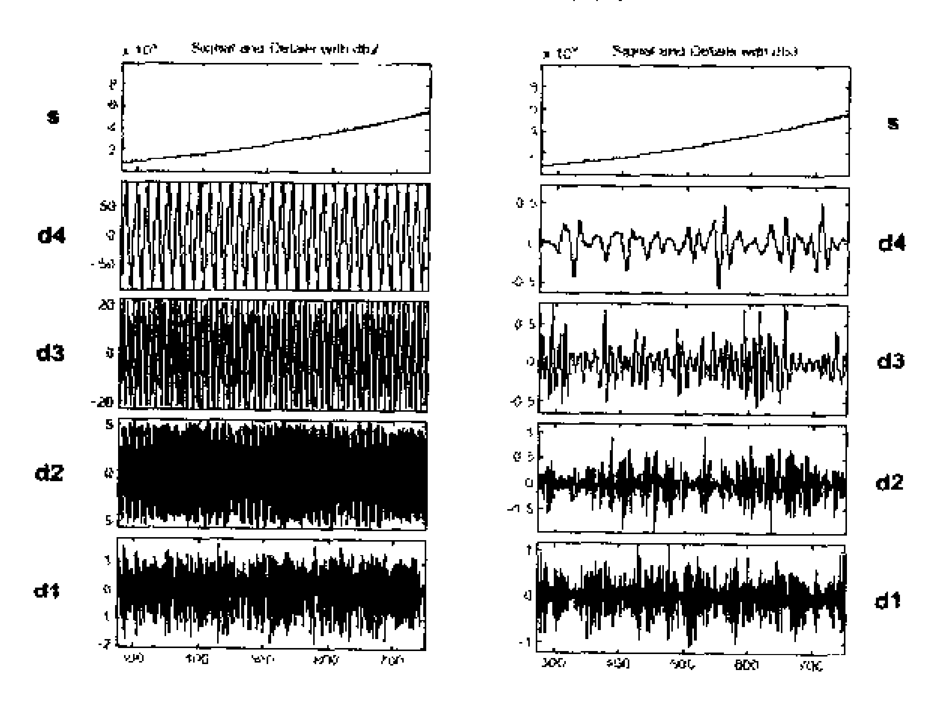


图 6-5 染噪多项式信号的小波分析

由于在这种信号情况下，小波分解的近似和信号本身的差别非常小，在此不予列出（有兴趣的读者可以通过 MATLAB 的 wavelet toolbox 查验）。由于 db2 小波随着分解层级的增加，其正则性增加的原因，它抑制了该多项式信号的零阶和一阶部分，而仅对该信号的二阶部分及噪声进行了分解，因此，我们发现在图的左列部分，除了细节 d1 中包含了该染噪信号的不规则性，其余各层细节中的信号周期性（规则性）随着层级的增加而增大。相反地，由于 db3 小波的正则性较差，所以它抑制了该信号的多项式部分，而析出了它的噪声部分。

应用小波分析可以较好地对该类信号进行抑制。

6.2.5 阶跃信号

应用小波对这个信号进行分析，是一个利用小波进行信号突变检测的最简单的示例。如图 6-6 所示，在小波分解的各个近似与细节层上都显示出了阶跃变化的时刻。

小波分析对阶跃信号的有效方面：

- 间断点的检测
- 信号抑制
- 信号未来发展趋势的检测
- 辨识细节和近似的变化范围

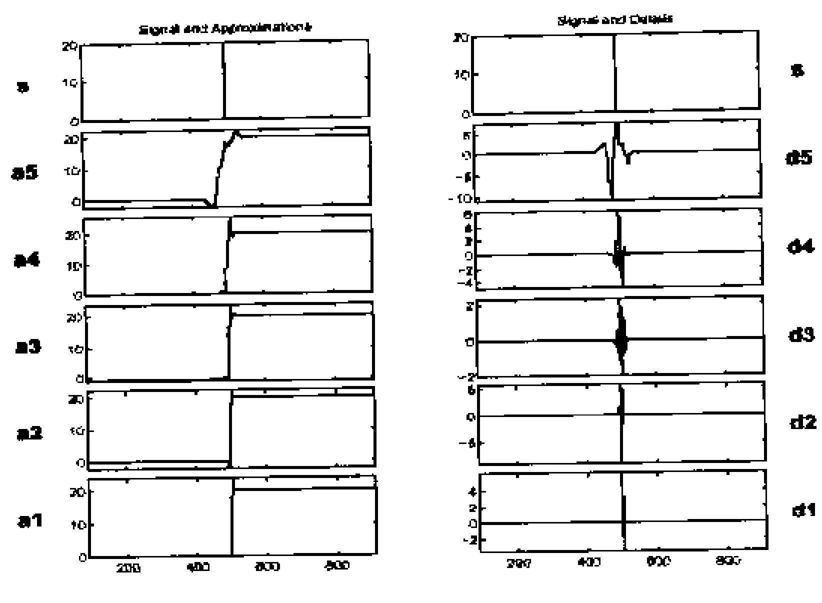


图 6-6　阶跃信号的小波分析

6.2.6　染噪的斜坡信号

如图 6-7 所示，信号 s 是一个被白噪声污染了的斜坡信号。利用 db3 小波对该信号进行 6 层分解，图 6-7 的左列显示了小波分解的近似。可以看出，随着噪声的减少，斜波的近似形式越来越精确。在第 6 层近似上已经显示了对斜波的较好重构。另外，虽然噪声影响了所有尺度上的分解结果，但是也能够较好地从染噪信号中分离出有用的斜坡信号。这是由于在较低层上的近似，使得噪声的影响能足够快地减小。

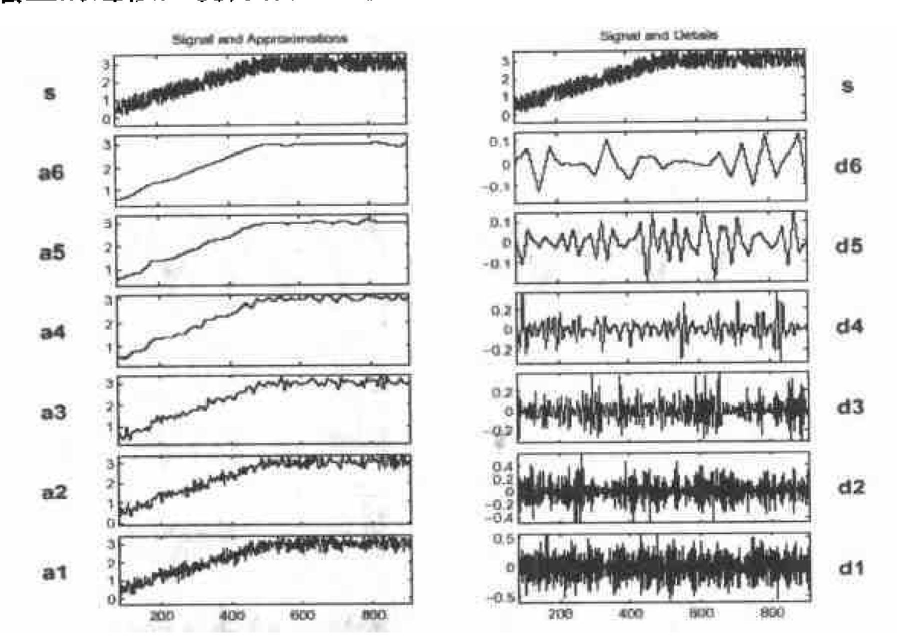


图 6-7　染噪（白噪声）斜坡信号的小波分解

图 6-8 给出了被有色噪声污染了的斜坡信号的小波分析结果。

小波分析对该类信号有效的分析领域有：

- 间断点的检测

- 噪声处理
- 信号未来发展趋势的检测
- 分离信号组分
- 辨识信号的近似与噪声
- 信号抑制

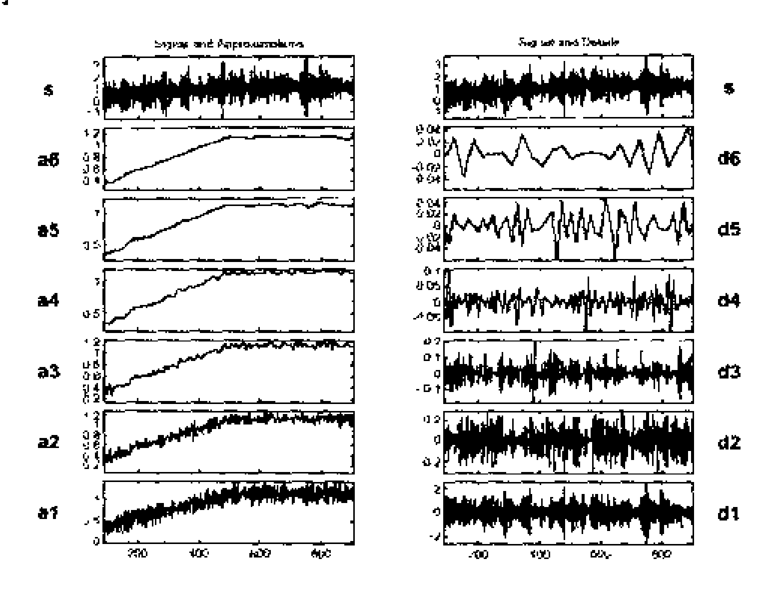


图 6-8　染噪（有色噪声）斜坡信号的小波分解

6.2.7　染噪的正弦信号

该信号是由一个正弦信号和白噪声信号叠加而成，如图 6-9 所示，应用 db5 小波对其进行 5 层分解。在这里，很好地体现了小波分解的线性特性：两个信号之和的分析等于两个信号分析的和，即相应的细节部分在白噪声的分解部分被表示出来了。

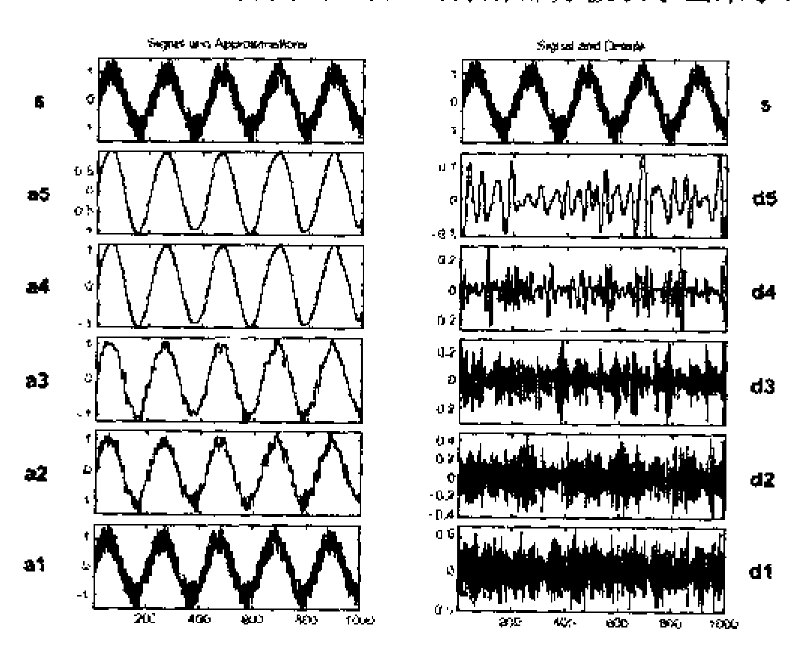


图 6-9　染噪正弦信号的小波分析

小波分析对该类信号有效的分析方面有：

- 噪声处理
- 信号未来发展趋势的检测
- 分离信号组分
- 辨识信号的频率

6.2.8　三角波与正弦波的合成信号

这里应用 db5 小波对一个由三角波和正弦波合成的信号进行小波分析，如图 6-10 所示。在图 6-10 的左列，近似 a6 清晰地显示出三角波的形状，在右列，细节 d1 和 d2 是非常小的，表明了在信号中没有包括那些相对于采样周期非常短的周期，细节 d3 特别是 d4 表示出了正弦波，细节 d6 则表示了边沿效应。

小波分析对该类信号有效的分析范围有：

- 信号未来发展趋势的检测
- 分离信号组分
- 辨识周期性信号的频率

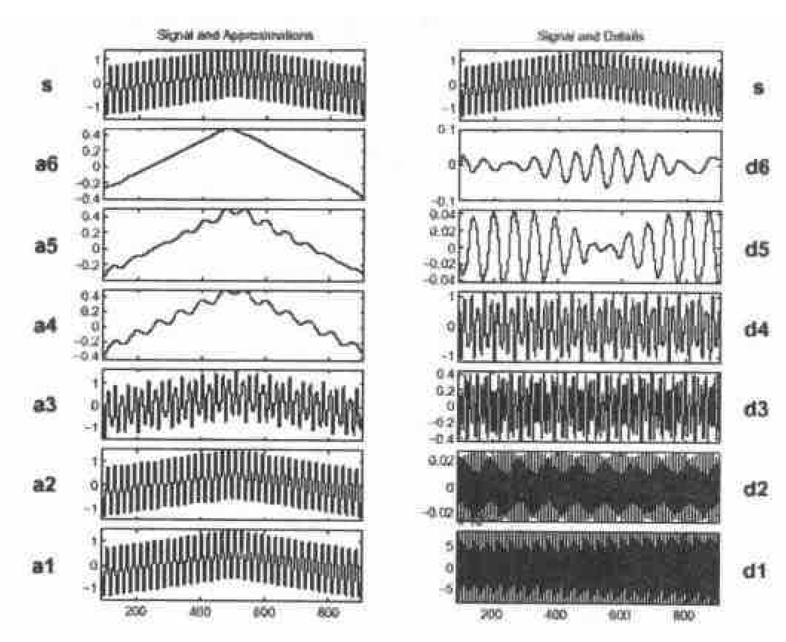


图 6-10　三角波与正弦波合成信号的小波分析

在此，仅对一些常用信号的某类小波分析结果做出了一定的分析。有兴趣的读者可以针对上述的各类信号，变换使用小波分析方法，观察所得到的结果，以加深小波分析在信号处理方面的印象。

6.3　信号的特征提取

在进行信号的分析时，如何提取信号的特征是一个关键的技术难点。信号的突变点往

往是它的重要特征。除此之外，信号的频率谱和它的幅值等表征了信号的许多信息。因此，在进行信号特征提取的研究中，信号的连续性（即信号的奇异性）分析、信号的频率谱分析和幅值分析等内容是不可或缺的。

应用小波分析进行信号特征提取时，主要有两种处理方法，即边界的处理和滤波。小波分析中，对边界的处理使用了延拓的方法，如对称延拓、周期延拓等。从理论上来说，对于有限区间$[a, b]$上的数据，应构造$L^2[a, b]$中的尺度函数与小波，从而得到$L^2[a, b]$上的小波基，这时的小波基往往带有边界条件。

在信号分析中，当对信号进行采样后，就得到在一个大的有限频带中的一个信号，对这个信号进行小波分解，其实质就是把采到的信号分成两个信号，即高频部分和低频部分，而低频部分通常包含了信号的主要信息。根据分析的需要，可以继续对所得到的低频部分进行再分解，如此又得到了更低频率部分的信号和频率相对较高部分的信号。当然，也可以对高频部分进行分解，不过这里用于分解的工具是小波包而已。这即是二进多尺度小波分解方法，这种方法把一个混频信号分解为若干个互不重叠的频带中的信号，这样就可以完成滤波或检波的工作，达到了提取信号特征的目的。

下面给出实例分析，以帮助读者进一步了解小波分析在信号处理方面的优越性。

【例 6-1】 检测一个信号的突变点。

这里所用信号为 MATLAB 软件中自带的信号，以下所用实例如不加特别说明，均同于此。

例程 6-1

```
%首先装载  原始信号
load nearbrk;
s=nearbrk;
===========================
%画出该信号的波形图
subplot(4,2,1);
plot(s);
Ylabel('s');
title('原始信号 s 和信号的近似 a、细节 d');
===========================
%用小波 db2 进行 3 层分解
[c,l]=wavedec(s,3,'db2');
for i=1:3   %此处的循环可使编程简捷
    decmpa=wrcoef('a',c,l,'db2',4-i);
    subplot(4,2,2*i+1);
    plot(decmpa);
    Ylabel(['a',num2str(4-i)]);
end
for i=1:3
    decmpd=wrcoef('d',c,l,'db2',4-i);
    subplot(4,2,2*(i+1));
    plot(decmpd);
    Ylabel(['d',num2str(4-i)]);
```

```
end
```

例程 6-1 运行结果如图 6-11 所示。

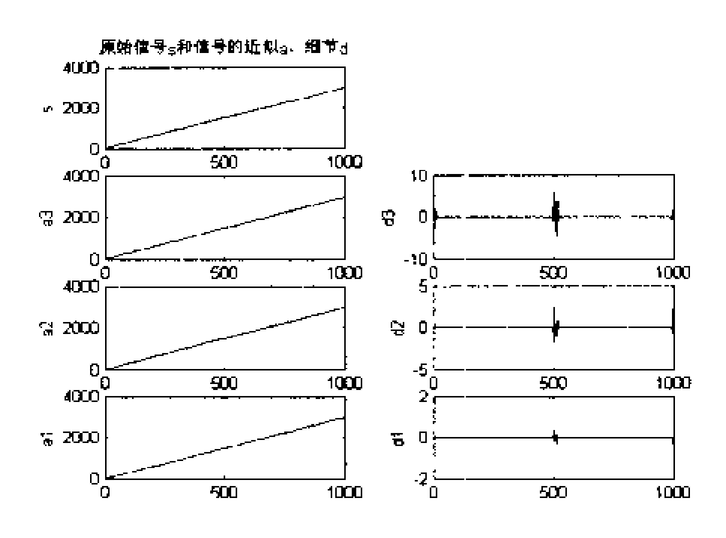


图 6-11 运行结果

在这个例子中，我们可以清晰地看到，对信号的小波分解将原信号的近似特征 a 和细节特征 d 提取出来了。在原始信号的图像上，无法得知原始信号的导数不连续性。

【例 6-2】给定一个正弦信号 $s(i)=\sin(i\times\pi/100+3\pi/4), i=0,1,\cdots,199$，对该小波进行多分辨分解与重构。

该例可说是一个纯数学问题，但是，它可以使读者清楚小波分析中多分辨分析在信号提取中的应用。

例程 6-2

```
t=0:pi/100:4*pi;
s=sin(t+3*pi/4);
plot(s);
title('原始信号 s');
======================================
%对 s 进行小波分解：db1   3 层
[c,l]=wavedec(s,3,'db1');
======================================
%提取小波分解的低频系数 a3
a3=appcoef(c,l,'db1',3);
======================================
%提取小波分解的各层高频系数
d3=detcoef(c,l,3);
d2=detcoef(c,l,2);
d1=detcoef(c,l,1);
======================================
%绘出各系数的图形
figure(2)
subplot(5,2,1);plot(a3);
```

```
Ylabel('a3');
subplot(5,2,3);plot(d3);
Ylabel('d3');
subplot(5,2,5);plot(d2);
Ylabel('d2');
subplot(5,2,7);plot(d1);
Ylabel('d1');
==========================================
%重构信号 s
s1=waverec(c,l,'db1');
subplot(5,2,9);plot(s1);
Ylabel('s1');
==========================================
%下面用小波 'coif3' 重复上述过程
[c,l]=wavedec(s,3,'coif3');
a3=appcoef(c,l,'coif3',3);
d3=detcoef(c,l,3);
d2=detcoef(c,l,2);
d1=detcoef(c,l,1);
subplot(5,2,2);plot(a3);
Ylabel('a3');
subplot(5,2,4);plot(d3);
Ylabel('d3');
subplot(5,2,6);plot(d2);
Ylabel('d2');
subplot(5,2,8);plot(d1);
Ylabel('d1');
s2=waverec(c,l,'coif3');
subplot(5,2,10);plot(s2);
Ylabel('s2');
```

例程 6-2 运行结果如图 6-12 所示。

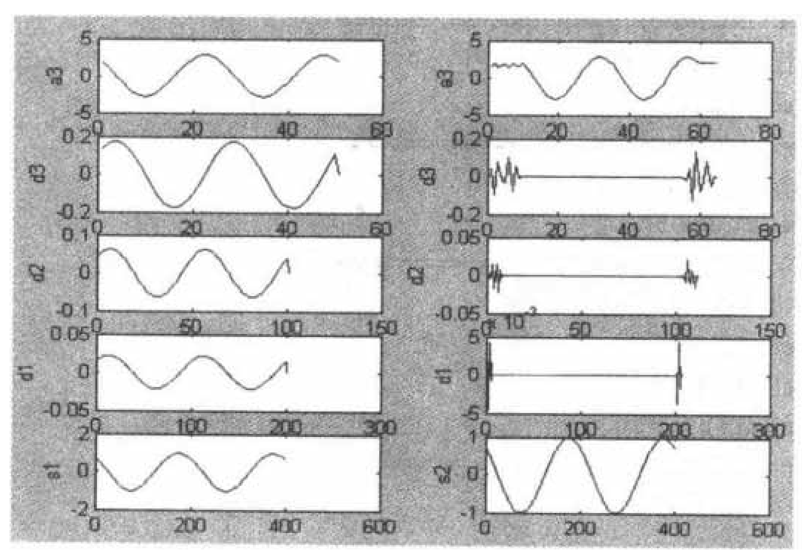


图 6-12 运行结果

6.4 信 号 处 理

6.4.1 信号的奇异性检测

信号中不规则的突变部分和奇异点往往包含有比较重要的信息，它是信号重要特征之一。在故障诊断中，例如，机械故障、电力系统故障、脑电图、心电图中的异常，以及地下目标的位置及形状等，都对应于测试信号的突变点，因而对突变点的检测在故障诊断中有着非常重要的意义。长期以来，傅里叶变换是研究函数奇异性的主要工具。但是傅里叶变换缺乏空间局部性，它只能确定一个函数奇异性的整体性质，而难以确定奇异点在空间的位置和分布情况。小波分析具有空间局部化性质，因此，利用小波分析来分析信号的奇异性及奇异性的位置和奇异度的大小是比较有效的。

这里有必要引入小波变换模极大值（或过零）点同信号突变点之间的关系。首先我们简单地介绍一下模极大值和一些相关的概念。

【定义】 在某一尺度 x_0 下，如果存在一点（x_0，y_0）使得 $\frac{\partial W_f(x_0,y_0)}{\partial y}=0$，则称点（$x_0$，$y_0$）是局部极值点，且 $\frac{\partial W_f(x_0,y)}{\partial y}$ 在 $y=y_0$ 上有一个**模极大值（过零）点**。如果对 y_0 的某一邻域内的任意点 y，有 $\left|W_f(x_0,y)\right|\leqslant\left|W_f(x_0,y_0)\right|$，则称（$x_0$，$y_0$）为小波变换**模极大值（过零）点**。尺度空间中所有的模极大值点的连线称为**模极大值线**。关于模极大值与信号的突变（奇异）点有下面的定理。

【定理】 设 n 为一严格的整数，Ψ 为具有 n 阶消失矩、n 次连续可微和紧支集的小波，$f(t)\in L^1(c,d)$（$[c,d]$为某一实数区间），若存在尺度 $x_0>0$，使得 $\forall x<0$，$t\in(c,d)$，$\left|W_f(x,y)\right|$没有局部极大值点，则在区间$(c+\varepsilon,d-\varepsilon)$上是一致 Lipschitz α（ε 为一任意小的正数）。一般来讲，函数在某一点的 Lipschitz 指数α表征了该点的奇异性大小，α越大，该点的光滑度越高；α越小，该点的奇异性越大。

我们知道，当小波函数可看做某一平滑函数的一阶导数时，信号小波变换模的局部极值点对应于信号的突变点（或边缘）；当小波函数可看做某一平滑函数的二阶导数时，信号小波变换模的过零点，也对应于信号的突变点（或边缘）。因此，采用检测小波变换系数模的过零点和局部极值点的方法可以检测信号的边缘位置。比较说来，用局部极值点进行检测更具优越性。

通常情况下，信号的奇异性可分为两种情况：一种是信号在某一时刻，起幅值发生突

变，引起信号的不连续，信号的突变处是第一种类型的间断点；另一种是信号外观上很光滑，其幅值没有突变，但是在信号的一阶微分上有突变产生，且一阶微分是不连续的，称此为第二种类型的间断点。

1. 第一种类型间断点的检测

【例 6-3】 现给定一个含突变点的信号，试用小波分析对该信号进行检测，判断其突变点的位置。

例程 6-3

```
load freqbrk;
s=freqbrk;
[c,l]=wavedec(s,6,'db5');
subplot(8,1,1);
plot(s);
title('使用 db5 小波分解 6 层：s=a6+d6+d5+d4+d3+d2+d1');
Ylabel('s');
a6=wrcoef('a',c,l,'db5',6);
subplot(8,1,2);
plot(a6);
Ylabel('a6');
d6=wrcoef('d',c,l,'db5',6);
subplot(8,1,3);
plot(d6);
Ylabel('d6');
d5=wrcoef('d',c,l,'db5',5);
subplot(8,1,4);
plot(d5);
Ylabel('d5');
d4=wrcoef('d',c,l,'db5',4);
subplot(8,1,5);
plot(d4);
Ylabel('d4');
d3=wrcoef('d',c,l,'db5',3);
subplot(8,1,6);
plot(d3);
Ylabel('d3');
d2=wrcoef('d',c,l,'db5',2);
subplot(8,1,7);
plot(d2);
Ylabel('d2');
d1=wrcoef('d',c,l,'db5',1);
subplot(8,1,8);
plot(d1);
Ylabel('d1');
```

例程 6-3 运行结果如图 6-13 所示。

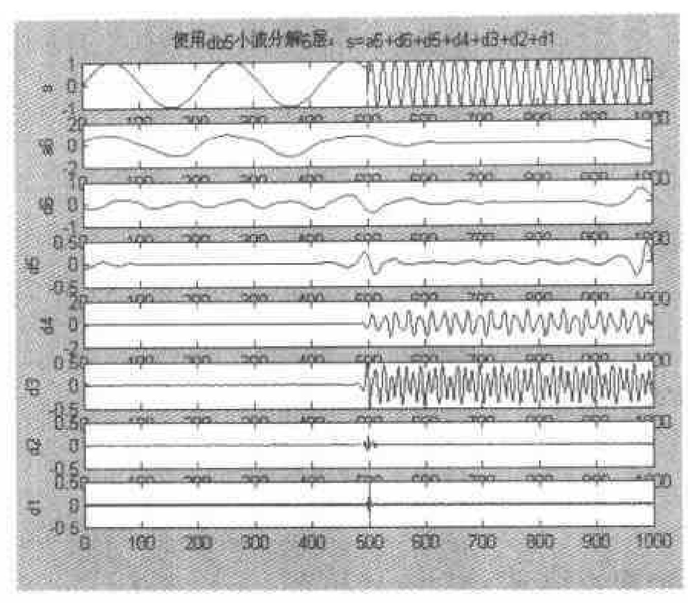


图 6-13　运行结果

在本例中，我们应用‘db5’小波将信号分解到第 6 层，来检测第一类型间断点，由图 6-13 可看出，在信号分解的细节部分清晰地显示出了间断点的准确位置。在这个例子中，信号的不连续性是由于在低频特征的正弦信号的后半部分，突然加入了具有中高频特征的正弦信号。在该信号的小波分解中，第 1 层和第 2 层细节（d1 和 d2）将信号的不连续性显示得相当明显，因为在该信号的断裂部分包含的是高频部分。如果只需辨别出信号的间断点，那么用 db1 小波比用 db5 小波的效果好。

2．第二种类型间断点的检测

【例 6-4】 对某一给定信号，请利用小波分析来检测出第二类间断点的准确位置。

例程 6-4

```
load nearbrk;
s=nearbrk;
[c,l]=wavedec(s,5,'db2');
subplot(7,1,1);
plot(s);
title('使用 db2 小波分解 5 层：s=a5+d5+d4+d3+d2+d1');
Ylabel('s');
a5=wrcoef('a',c,l,'db2',5);
subplot(7,1,2);
plot(a5);
Ylabel('a5');
d5=wrcoef('d',c,l,'db2',5);
subplot(7,1,3);
plot(d5);
Ylabel('d5');
d4=wrcoef('d',c,l,'db2',4);
subplot(7,1,4);
plot(d4);
```

```
Ylabel('d4');
d3=wrcoef('d',c,l,'db2',3);
subplot(7,1,5);
plot(d3);
Ylabel('d3');
d2=wrcoef('d',c,l,'db2',2);
subplot(7,1,6);
plot(d2);
Ylabel('d2');
d1=wrcoef('d',c,l,'db2',1);
subplot(7,1,7);
plot(d1);
Ylabel('d1');
```

例程 6-4 运行结果如图 6-14 所示。

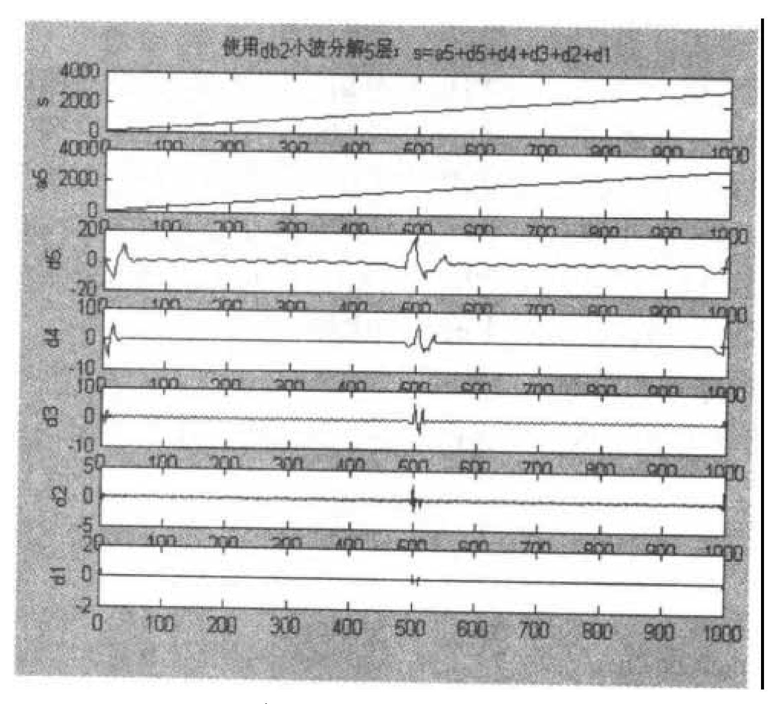


图 6-14　运行结果

在这个例子中，外观上，原始信号是一条光滑的直线，但是它的一阶微分有突变。我们所利用的 db2 小波对信号进行分解后，便将该信号的第二种类型间断点显现出来了。在此，应特别注意，检测信号的第二种类型间断点时，所用分析小波的正则性是非常重要的，如果选择了不具有正则性的小波进行分析，将检测不出来第二种类型间断点。

在本节中所涉及到的信号间断点的检测问题，在图像（可看做是二维信号）处理中，一个主要的问题就是边缘检测，也包括了突变的检测。依据本节所介绍的方法，也可以找到动态系统中具有非常迅速发展的瞬态信号。而这些现象所具有的主要特征就是在时间或者空间上发生的局部变化，进行分析的目的则是决定如下几点：

- 变化的场所（如时间或空间）
- 变化的类型（信号本身或其一阶、二阶导数的突变）

- 变化的幅度

在检测信号突变时，短的小波通常比长的小波更有效。通过最小的小波辨识不连续性的形状比用最长的小波做同样的事情更为简单。因此：

- 辨识信号的不连续性，使用 haar 小波就足够了；
- 辨识 j 阶导数的不连续性，需选择至少具有 j 阶消失矩的正则小波。

6.4.2　信号自相似性的检测

在进行分析之前，先介绍一下小波系数与自相似性的关系。直观上说，小波分解可通过计算信号和小波之间的“自相似指数”得到。这里的自相似指数也就是小波系数，如果自相似指数大，则信号的自相似程度就高，反之亦然。如果一个信号在不同的尺度上都相似于它本身，那么，其“自相似指数”，或者小波系数在不同的尺度上也是相似的。在此，通过例子来说明小波分析是如何检测信号的自相似性的。

【例 6-5】 请利用小波分析来检测一个给定信号的自相似性。这里所用信号为一个经过反复迭代生成的合成信号。

例程 6-5

```
load vonkoch;
s=vonkoch;
subplot(2,1,1);
plot(s);
title('原始信号');
subplot(2,1,2);
f=cwt(s,[2:2:128],'coif3','plot');
title('小波分解自相似指数图');
Xlabel('时间');
Ylabel('变换尺度');
```

例程 6-5 运行结果如图 6-15 所示。

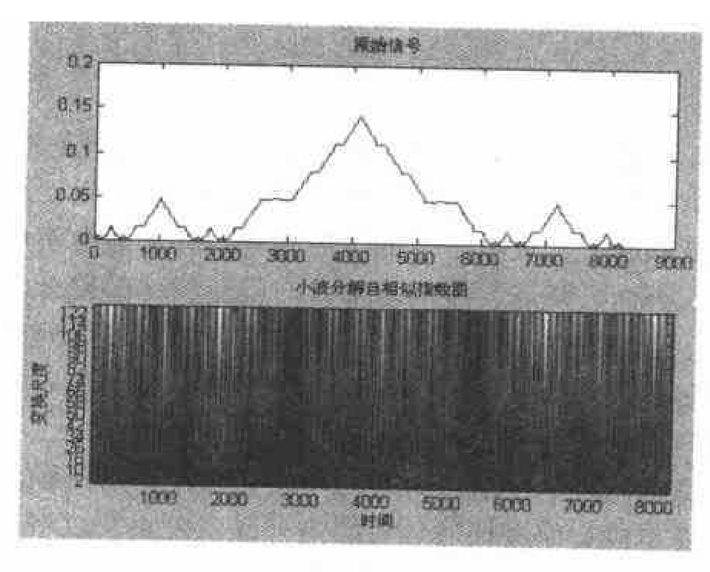


图 6-15　运行结果

在小波分解后显示的自相似指数图中，会发现在许多尺度上，小波系数看上去都是很相似的，图 6-15 中，垂直轴线上显示的线条就是由于信号的自相似性产生的。小波系数越大，则灰度越深。

由于信号的自相似性也即是信号的分形特征。目前，进行该项研究的人员很多，研究结果表明，采用小波分解，可以很好地研究信号或图像的分形特征。在开始，分形特征随着时间的发展而变化，随后又不随时间的发展而发生变化，称这种信号为分形（miltifractal）。实践表明，小波分析工具非常适合于分形的实际研究和分形的生成。

6.4.3 信号发展趋势的识别

通常，一些染噪信号的发展趋势是难以分辨的。由于噪声的污染，对我们有用的信号的发展趋势在时域中看不出来，但是，通过小波分解，可以去除那些干扰信号，最终显现出有用信号的真面目。

【例 6-6】现有一个被有色噪声所污染了的斜坡信号，请利用小波分析出这个信号的发展趋势。

例程 6-6

```
load cnoislop;
s=cnoislop;
subplot(7,1,1);
plot(s);
Ylabel('s');
title('原始信号和各层近似');
[c,l]=wavedec(s,6,'db3');
for i=1:6
    decom=wrcoef('a',c,l,'db3',7.i);
    subplot(7,1,i+1);
    plot(decom);
    Ylabel(['a',num2str(7.i)]);
end
```

结果如图 6-16 所示。

从原始信号 s 中可看出，由于噪声的污染，信号的发展趋势是不可见的。而在本例中，利用 db3 小波分解到第 6 层，则信号的发展趋势随着近似的变化（从 a1 到 a6），而变得越来越清晰。信号中的低频部分代表着信号的发展趋势，在小波分析中，则对应着最大尺度小波变换的低频系数。因而，随着尺度的增加，时间分辨率的降低，信号的发展趋势会表现得更为明显。此外，还可以在频率中理解它的含义，即尺度分解中的低频部分随着层次的增加，它所含有的高频信息会随之减小。当分解到下一个层次时，就有更高一些的频率信息被滤掉，而所剩下的就是信号的发展趋势。

在此可看出，在展示信号的发展趋势时，小波分析是很有用的。这种分析所具有的另一个目的是将隐藏在噪声或其他高频信号中的信号显示出来。这里需要强调一点，所有来辨识的信号本身不能具有很大的突变，这是因为信号的发展趋势是由信号的低频部分所表

征的。如果在信号本身中包含有很大的突变，那么在多尺度小波变换的低频部分中，显示出来的信号会和原始信号有很大的差别，因为这种变换将信号本身的突变当做高频信息给滤掉了。

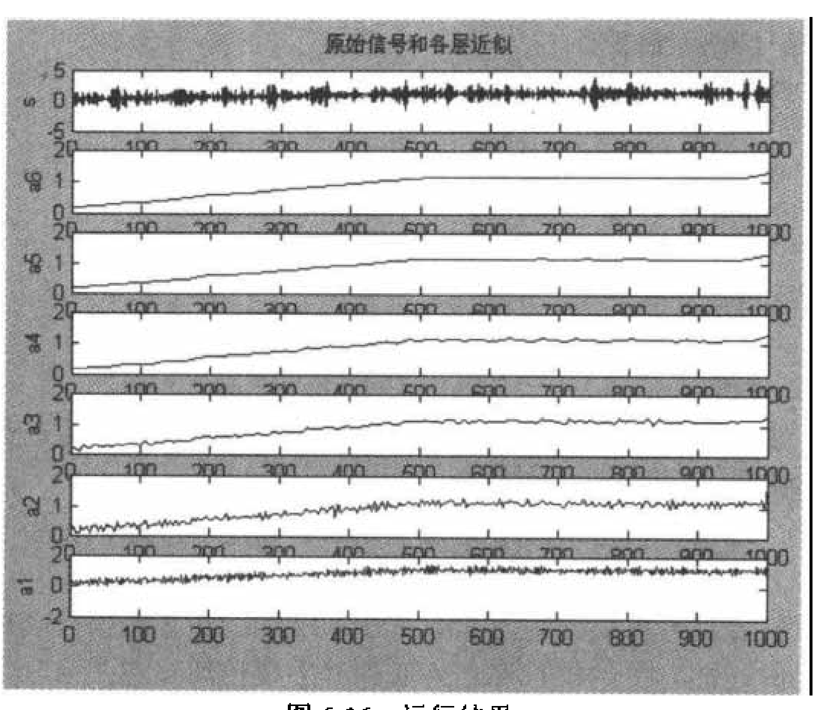


图 6-16　运行结果

6.4.4　某一频率区间上信号的识别

【例 6-7】请利用小波分析将一给定信号的各频率成分区分开。

这里所用于分析的信号是由三个不同频率的正弦信号所组成的，我们用 db3 小波实现这个处理。

例程 6-7

```
load sumsin;
s=sumsin;
figure(1);
subplot(6,1,1);
plot(s);
Ylabel('s');
title('原始信号和各层近似');
[c,l]=wavedec(s,5,'db3');
for i=1:5
    decom=wrcoef('a',c,l,'db3',6-i);
    subplot(6,1,i+1);
    plot(decom);
    Ylabel(['a',num2str(6-i)]);
end
```

```
figure(2);
subplot(6,1,1);
plot(s);
Ylabel('s');
title('原始信号和各层细节');
[c,l]=wavedec(s,5,'db3');
for i=1:5
    decom=wrcoef('d',c,l,'db3',6.i);
    subplot(6,1,i+1);
    plot(decom);
    Ylabel(['d',num2str(6.i)]);
end
```

结果如图 6-17 所示。

在图 6-17 所显示的分解结果中可以看到，在近似的第四层中将正弦信号中的最低频率组成清晰地显示出来了。

在小波分解中，通常将信号中的最高频率成分看做是 1，那么各层小波分解便是带通或低通滤波器。

在本例中，该信号是由周期分别为 200、20 和 2 的三个正弦信号合成的，它们的采样周期均为 1，为了叙述的方便，权且称之为低频、中频和高频正弦信号。由信号频率的组成和小波分解后各层频率的分布可知，高频信号位于细节 d1 层，从图中可看出，每一个信号中含有 10 个正弦振荡。而在细节 d4 层包含中频正弦信号。同时注意到，在近似 a3 和 a4 层之间出现了信号的不连续性，因为中频信号是由这两层共同表达的，d4 层的信息被 d3 层截去了一部分。因此我们需要用近似 a1 到 a3 的信息估计中频正弦信号。如果将 a1 放大就会看出周期约为 20 的中频正弦信号，而周期约为 2 的低频信号在近似 a4 层中可以清晰地分辨。

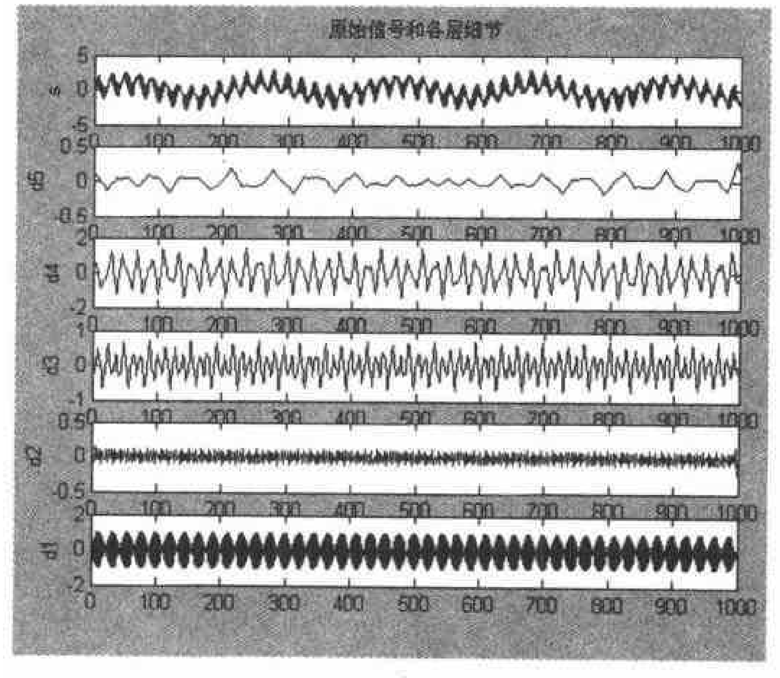


图 6-17　运行结果

总之，由于在小波分解下，不同的尺度具有不同的时间和频率分辨率，因而利用小波分解可以将信号的不同频率区间所包含的信号分离出来。

6.4.5　信号抑制与衰减

在介绍信号的抑制与衰减之前，先介绍一下有关消失矩的概念。

【定义】简单地叙述即是，如果 $x^k\Psi(x)$（其中 $\Psi(x)$ 是小波函数）（k=0,1,2,⋯,n）的平均值为 0，那么该小波有 n+1 个消失矩，并且可以利用该小波对 n 次多项式信号进行抑制。

下面先通过一个例子来阐述应用小波分析进行信号的抑制与衰减。

【例 6-8】　请利用小波分析对一给定信号的高频部分进行抑制。

例程 6-8

```
%装载原始信号
load sumsin;
s=sumsin;
==============================
%设置小波名并利用 coif3 小波进行 4 层分解
w='coif3';
maxlev=4;
[c,l]=wavedec(s,maxlev,w);
newc=c;
==============================
%将分解后的第三、四层细节系数置为 0
newc=wthcoef('d',c,l,[3,4]);
==============================
%在原始信号的时间区间[400，600]内将第一层细节系数置为 0
%并且将其他系数进行衰减，求出第一层系数起始点和终止点的
%索引值
k=maxlev+1;
first=sum(l(1:k.1))+1;
last=first+l(k).1;
indd1=first:last;
==============================
%将系数除以 3，进行信号衰减
newc(indd1)=c(indd1)/3;
==============================
%在区间[400，600]上求出第一层系数索引
indd1=(first+400/2):(first+600/2);
==============================
%将该索引值置为 0
newc(indd1)=zeros(size(indd1));
```

```
%=============================
%将第二层中相应于原始信号 t=500 的时间点处的系数置为 4
k=maxlev;
first=sum(l(1:k.1))+1;
newc(first+500/2^2)=4;
%=============================
%综合修改后的分解结构
synth=waverec(newc,l,w);
%=============================
%用图示出上述修改结果
subplot(2,2,1);
plot(s);
title('原始信号');
subplot(2,2,2);
plot(c);
title('coif3 小波分解后的系数');
subplot(2,2,3);
plot(synth);
title('小波抑制后的信号');
subplot(2,2,4);
plot(newc);
title('修改后的小波分解系数');
```

结果如图 6-18 所示。

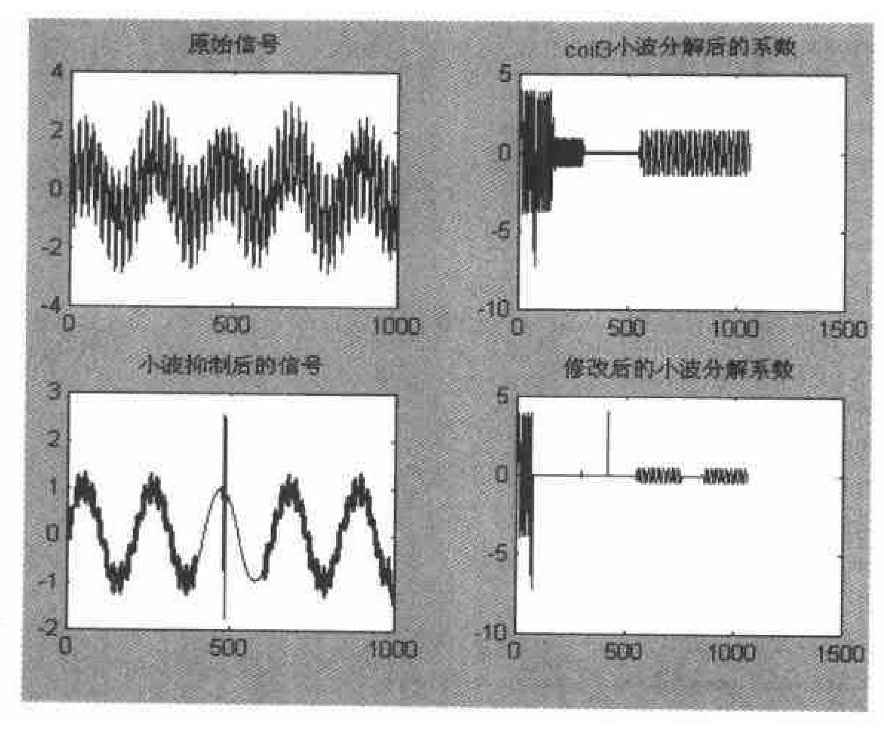


图 6-18　运行结果

通过上例可以知道，对一个多项式用小波进行分解时，可以使信号分解的高频部分变成 0，这是因为小波的过零点的个数超过了多项式的次数。

利用小波分析进行信号抑制是小波分析应用于实际的重要方面之一。一般地，可以用 k 次多项式逼近一个信号。所以，小波对信号的抑制可归结为对多项式幅值的抑制，而这种抑制的能力取决于小波本身的一个重要的数学特征，即**过零点**。这里可以将过零点理解为一种平均值的外延。如果一个小波的平均值为 0，那么它至少有一个过零点。在此，过零点的概念就是前面所提到的消失矩的概念。

如果小波函数 Ψ 是一个至少具有 $n+1$ 个过零点（即对 $j=0,1,\cdots,k$，有 $\int_R x^j\Psi(x)\mathrm{d}x=0$），且信号 s 是一个 k 次多项式，则系数 $C(a,b)=0$ 对所有的 a 和 b 都成立，那么这种小波函数就可以自动地抑制多项式信号，此时，信号 s 的次数可随时间变化，但必须保证信号 s 的次数小于 k。

如果信号 s 是一个在区间 $[\alpha,\beta]$ 内的 k 次多项式，那么只要函数 $\frac{1}{\sqrt{a}}\Psi\left(\frac{x-b}{a}\right)$ 在区间 $[\alpha,\beta]$ 内成立，且系数 $C(a,b)=0$ 也成立，则小波分析就可对该信号进行抑制，但是在信号的边界部分会有衰减作用。因此，可以在区间 $[\alpha,\beta]$ 中，假设信号 s 可展开成形式为 $s(x)=s(0)+xs(0)+x^2s^{(2)}(0)+\cdots+x^ks^{(k)}(0)+g(x)$，则 s 和 g 具有相同的小波系数。在此可以用相位的观点来理解：小波仅抑制了信号 s 中的多项式部分，而对信号中的不规则部分 g 不起作用。因此小波分析是通过对规则信号的抑制来达到分析不规则信号的目的。

抑制信号中某些成分的另一种方法是将小波分解系数中的某些系数 $C(a,b)$ 强制性地置为 0。对于一个系数组成的集合 F，有 $\forall(a,b)\in F$，令 $C(a,b)=0$，再将修改后的小波分解系数进行小波重构，即可对该信号的某些部分进行抑制。上例应用了这种方法。

下面我们再通过一个染噪的二次多项式来说明小波分析对信号的抑制作用。

【例 6-9】 请利用小波分析对一给定信号进行某些频率上的衰减。

例程 6-9

```
load noispol;
s=noispol;
=======================
%用 db3 小波进行 4 层分解
[c,l]=wavedec(s,4,'db3');
subplot(5,1,1);
plot(s);
```

```
title('原始信号及各层细节信号重构图');
Ylabel('s');
==========================
%重构分解结构[c，l]中的高频部分
for i=1:4
    decmp=wrcoef('d',c,l,'db3',5-i);
    subplot(5,1,i+1);
    plot(decmp);
    Ylabel(['d',num2str(5-i)]);
end
```

结果如图 6-19 所示。

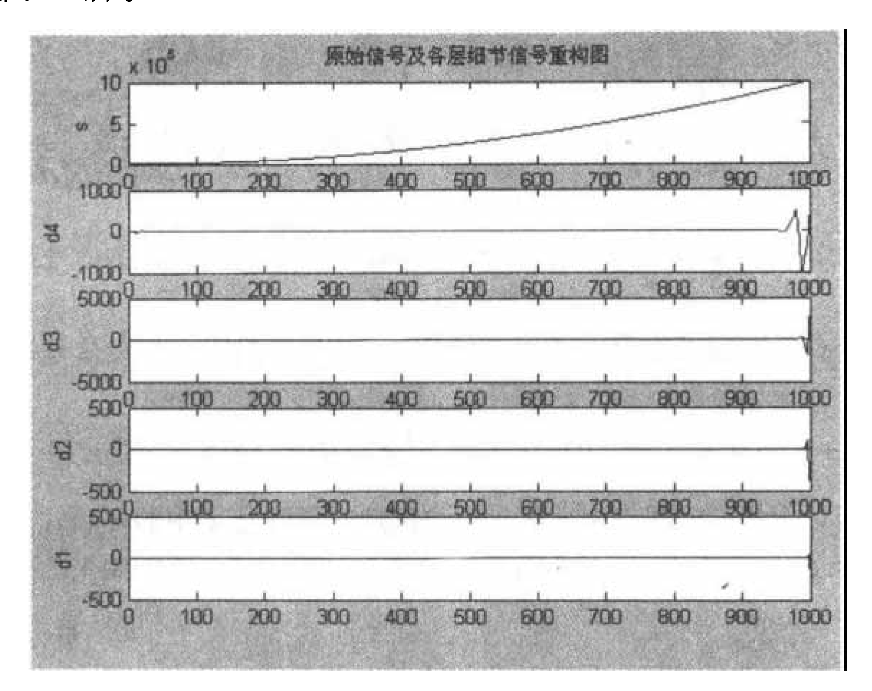


图 6-19 运行结果

从图 6-19 中可以看出，几乎所有细节的小波系数都为 0，这是因为所用的 db3 小波有三个消失矩，因而使得各尺度的高频系数几乎为 0。

6.4.6 信号消噪与提取弱信号

小波分析的重要应用之一就是用于信号消噪。在此，我们简要地阐述一下小波分析对信号消噪的基本原理。

我们知道，一个含噪的一维信号模型可表示为如下形式：

$$s(k) = f(k) + \varepsilon \cdot e(k), k = 0,1,\cdots,n-1$$

其中，$s(k)$ 为含噪信号，$f(k)$ 为有用信号，$e(k)$ 为噪声信号。这里我们认为 $e(k)$ 是一个 1 级高斯白噪声，通常表现为高频信号，而工程实际中 $f(k)$ 通常为低频信号，或

者是一些比较平稳的信号。因此我们可按如下的方法进行消噪处理：首先对信号进行小波分解，一般地，噪声信号多包含在具有较高频率的细节中，从而可利用门限阈值等形式对所分解的小波系数进行处理，然后对信号进行小波重构即可达到对信号的消噪之目的。对信号消噪实质上是抑制信号中的无用部分，恢复信号中有用部分的过程。

一般地，一维信号消噪的过程可分为如下3个步骤：

（1）一维信号的小波分解。选择一个小波并确定分解的层次，然后进行分解计算；

（2）小波分解高频系数的阈值量化。对各个分解尺度下的高频系数选择一个阈值进行软阈值量化处理。

（3）一维小波重构。根据小波分解的最底层低频系数和各层高频系数进行一维小波重构。

这3个步骤中，最关键的是如何选择阈值及如何进行阈值量化，在某种程度上，它关系到信号消噪的质量。

1. 噪声在小波分解下的特性

在此，我们将噪声看做普通信号分析一下它的相关性、频谱和频率分布这3个主要特征。

总体上，对于一维离散信号来说，其高频部分所影响的是小波分解的第一层细节，其低频部分所影响的是小波分解的最深层和低频层。如果对一个仅由白噪声所组成的信号进行分析，则可得出这样的结论：高频系数的幅值随着分解层次的增加而迅速地衰减，且其方差也有同样的变化趋势。在这里用$C(j,k)$表示对噪声用小波分解后的系数，其中，j表示尺度，k表示时间，对离散时间信号引入如下的属性：

（1）如果所分析的信号s是一个平稳、零均值的白噪声，那么它的小波分解系数是相互独立的。

（2）如果信号s是一个高斯型噪声，那么其小波分解系数是互不相关的，且服从高斯分布。

（3）如果信号s是一个平稳、有色、零均值的高斯型噪声序列，那么它的小波分解系数也是高斯序列。并且对每一个分解尺度j，其相应的系数是一个平稳、有色的序列。如何选择对分解系数具有解相关性的小波是一个很困难的问题，在目前也没有得到很好地解决。进一步需指出，即使存在一个小波，但是它对噪声的解相关性取决于噪声的有色性，为了用小波计算噪声的解相关性，必须知道噪声本身的颜色。在此不加详述，有兴趣的读者可参考有关资料。

（4）如果信号s是一个固定的零均值ARMA模型，那么对每一个小波分解尺度j，$C(j,k)(k \in z)$也是固定的零均值ARMA模型，且其特性取决于尺度j。

（5）如果信号s是一个噪声：

- 若它的相关函数已知，则可计算系数序列$C(j,k)$和$C(j,k')$；
- 若它的相关函数谱已知，则可计算$C(j,k)(k \in z)$的谱及尺度j和j'的交叉谱。

2. 应用一维小波分析进行信号消噪处理

这里我们先介绍两个非常有用的函数 wden 和 wdencmp。

信号消噪的主要函数 wden 的最简单的用法如下：

```
sd=wden(s, tptr, sorh, scal, n, wavename)
```

它所返回的是经过对原始信号 s 进行消噪处理后的信号 sd。其中，tptr 指定阈值选取规则，sorh 指定选取软阈值（sorh='s'）或硬阈值（sorh='h'），n 为小波分解的层数，wavename 指定分解时所用的小波。scal 是阈值尺度改变的比例，它有如下 3 种选择：

（1）scal='one'，表示基本模式。

一般地，可以忽略必须估计的噪声层次。在小波分解的细节层 cD1（最精细的尺度）中，主要包含的是噪声系数，其标准偏差等于σ，而绝对标准偏差比较稳定，其估计值总等于σ。这种稳定的估计值在信号分析中是相当重要的。这是因为，一方面，如果第一层的系数中含有有用信号的高频信息，且其本身是很规则的，那么这种高频信息在几个高频层中能够集中地显现出来；另一方面，可以避免信号本身的截短效应，折中截短效应是在计算信号的边缘时所产生的无用信息。当这个无用信息被认为是一个非白噪声时，则必须在每个不同的小波分解尺度上估计噪声的层次，并以此来变换阈值尺度。

（2）scal='sln'，表示未知尺度的基本模式，且仅根据第一层的小波分解系数来估计噪声的层次，并只进行一次估计，以此来变换阈值的尺度。

（3）scal='mln'，表示非白噪声的基本模式，且在每个不同的小波分解层次上都估计噪声的层次，以此来变换阈值的尺度。

wdencmp 是一种用得更为普遍的函数，它可以直接对一维或二维信号进行消噪或压缩，处理方法也是通过对小波分解系数进行阈值量化来实现。其使用方法如下：

```
xd=wdencmp(opt, x, wavename, n, thr, sorh, keepapp)
```

其中：

（1）opt='gbl'，thr>0，则阈值为全局阈值；
　　opt='lvd'，thr 是向量，则阈值是在各层上大小不同的数值。

（2）keepapp=1，不对小波分解后的低频系数做任何处理；
　　keepapp=0，对小波分解后的低频系数也进行阈值量化处理。

（3）x 是待处理信号。

（4）xd 是处理后信号。其余参数同函数 wden 中的参数。

小波分析进行消噪处理一般有下述 3 种方法：

（1）默认阈值消噪处理。该方法利用函数 ddencmp 生成信号的默认阈值，然后利用函数 wdencmp 进行消噪处理。

（2）给定阈值消噪处理。在实际的消噪处理过程中，阈值往往可通过经验公式获得，且这种阈值比默认阈值的可信度高。在进行阈值量化处理时可用函数 wthresh。

（3）强制消噪处理。该方法是将小波分解结构中的高频系数全部置为 0，即滤掉所有高频部分，然后对信号进行小波重构。这种方法比较简单，且消噪后的信号比较平滑，但是容易丢失信号中的有用成分。

下面用具体的例子来说明小波分析对信号的消噪作用。

【例 6-10】请利用小波分析对染噪信号进行消噪。

例程 6-10

```
%给定一个正弦信号并图示之
t=0:1000;
s=sin(0.03*t);
subplot(3,1,1);
plot(s);
axis([0 1000 .1 1]);
title('原始信号');
==============================
%给该信号加噪声
load noissin;
ns=noissin;
subplot(3,1,2);
plot(ns);
title('染噪信号');
==============================
%进行消噪处理
xd=wden(ns,'minimaxi','s','one',5,'db3');
subplot(3,1,3);
plot(xd);
title('消噪信号');
```

结果如图 6-20 所示。

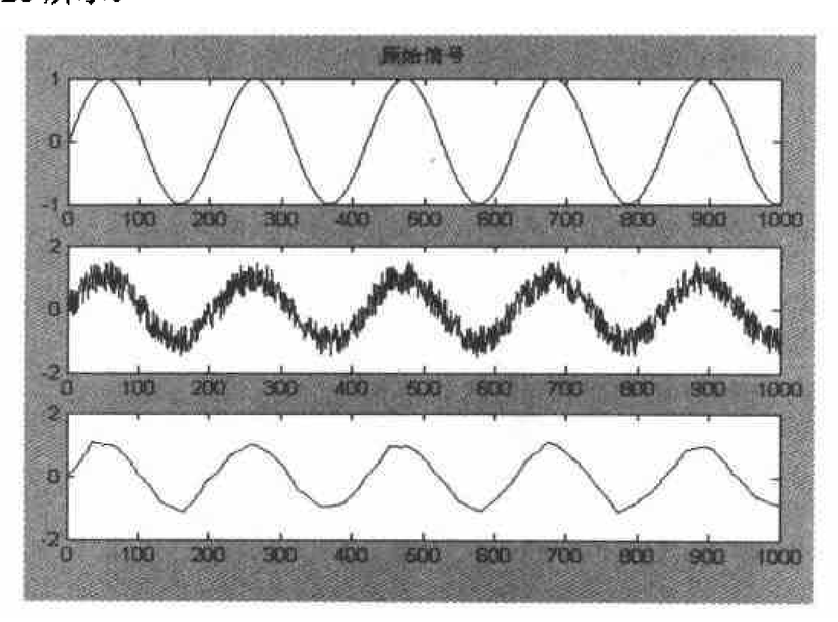


图 6-20　运行结果

在这个例子中，展示了从噪声数据中恢复信号的问题。从图中可以看出，消噪后的信号大体上显示了原始信号的形状，并明显地除去了噪声所引起的干扰。但是，所恢复的信号和原始信号相比，有明显的改变。这主要是因为在进行消噪处理的过程中所用的分析小波和细节系数阈值不恰当所致。下面我们再看一个例子。

【例 6-11】对某地的用电情况进行考察，对其电网电压值进行监测。在采样过程中，监测设备出现了一点故障，致使所采集到的信号受到噪声的污染。现在利用小波分析，对污染信号进行消噪处理以恢复原始信号。

例程 6-11

```
%装载采集的信号 leleccum.mat
load leleccum;
============================
%将信号中第 2000 到第 3450 个采样点赋给 s
indx=2000:3450;
s=leleccum(indx);
============================
%画出原始信号
subplot(2,2,1);
plot(s);
title('原始信号');
============================
%用 db1 小波对原始信号进行 3 层分解并提取系数
[c,l]=wavedec(s,3,'db1');
a3=appcoef(c,l,'db1',3);
d3=detcoef(c,l,3);
d2=detcoef(c,l,2);
d1=detcoef(c,l,1);
============================
%对信号进行强制性消噪处理并图示结果
dd3=zeros(1,length(d3));
dd2=zeros(1,length(d2));
dd1=zeros(1,length(d1));
c1=[a3 dd3 dd2 dd1];
s1=waverec(c1,l,'db1');
subplot(2,2,2);
plot(s1);grid;
title('强制消噪后的信号');
============================
%用默认阈值对信号进行消噪处理并图示结果
%用 ddencmp 函数获得信号的默认阈值
[thr,sorh,keepapp]=ddencmp('den','wv',s);
s2=wdencmp('gbl',c,l,'db1',3,thr,sorh,keepapp);
subplot(2,2,3);
plot(s2);grid;
title('默认阈值消噪后的信号');
============================
%用给定的软阈值进行消噪处理
softd1=wthresh(d1,'s',1.465);
softd2=wthresh(d2,'s',1.823);
softd3=wthresh(d3,'s',2.768);
```

```
c2=[a3 softd3 softd2 softd1];
s3=waverec(c2,l,'db1');
subplot(2,2,4);
plot(s3);grid;
title('给定软阈值消噪后的信号');
```

结果如图 6-21 所示。

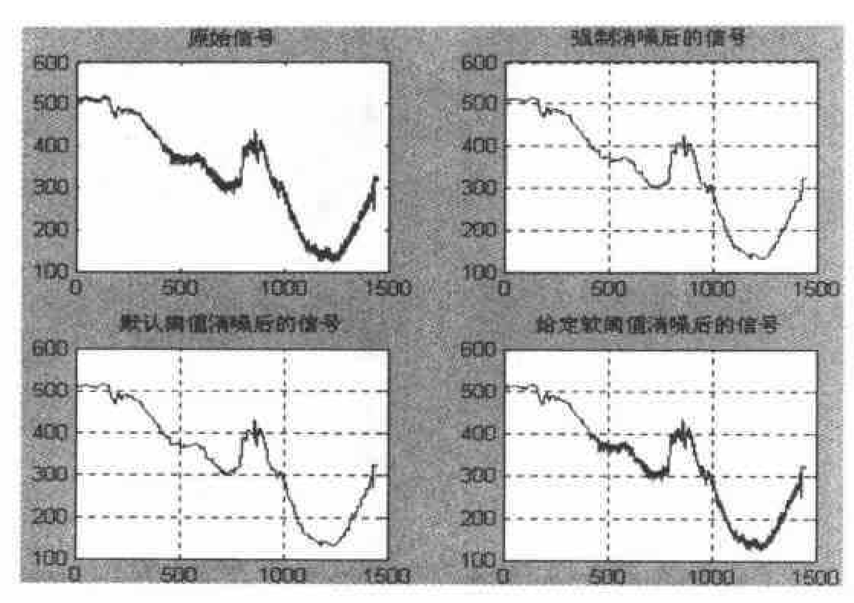


图 6-21　运行结果

在本例中，我们分别用前面所提到的三种消噪方法进行处理。从所得到的结果来看，应用强制消噪处理后的信号较为光滑，但是它很有可能丢失了信号中的一些有用成分。而默认阈值消噪和给定软阈值消噪这两种处理方法在实际中应用得更为广泛一些。

3．**阈值选取规则**

在 MATLAB 的小波工具箱中，提供了一个生成阈值的函数：

yt=wthresh(y，sorh，thr)

该函数根据参数 sorh 的取值返回输入分解系数的软阈值或硬阈值。其中，硬阈值对应于最简单的处理方法，而软阈值具有很好的数学特性，并且所得到的理论结果是可用的。

【例 6-12】生成阈值。

例程 6-12

```
y=linspace(.1,1,100);
thr=0.28;
ythard=wthresh(y,'h',thr);
ytsoft=wthresh(y,'s',thr);
subplot(2,2,1);
plot(y);
title('原始信号');
subplot(2,2,3);
```

```
plot(ythard);
title('硬阈值信号');
subplot(2,2,4);
plot(ytsoft);
title('软阈值信号');
```

结果如图 6-22 所示。

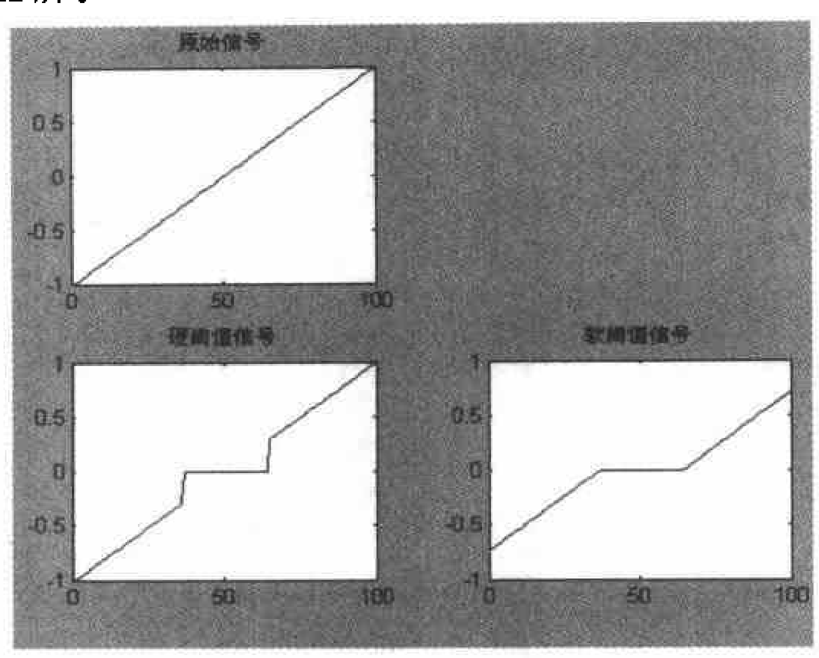


图 6-22　运行结果



令 t 表示阈值，则硬阈值信号 s 的形式为：$s=\begin{cases} x & |x|>t \\ 0 & |x|\leqslant t \end{cases}$

软阈值信号 s 的形式为：$s=\begin{cases} sign(x)(|x|-t) & |x|>t \\ 0 & |x|\leqslant t \end{cases}$

下面介绍在进行信号消噪处理时选取阈值的一般规则。

根据基本的噪声模型，选取阈值时有如下四个规则，其中每一条规则对应于函数 thselect 中输入参数 tptr 的一个选项：

- 选项 tptr='rigrsure'是一种基于 Stein 的无偏似然估计原理的自适应阈值选择。给定一个阈值 t，得到它的似然估计，再将非似然 t 最小化，就可得到所选的阈值。这是一种软件阈值估计器。
- 选项 tptr='sqtwolog'所采用的是一种固定的阈值形式，它所产生的阈值为 sprt(2*log(length(X)))。
- 选项 tptr='heursure'是前两种阈值的综合，所选择的是最优预测变量阈值。如果信噪比很小，而 SURE 估计有很大的噪声，在这种情况下，就需采用这种固定的阈值形式。
- 选项 tptr='minimaxi'也是一种固定的阈值选择形式，它所产生的是一个最小均方差的极值，而不是无误差。在统计学上，这种极值原理常用来设计估计器。因为

被消噪的信号可以看做是与未知回归函数的估计式相似，这种极值估计器可在给定的函数中实现最大均方误差最小化。

如果 y 是一个高斯白噪声信号 $N(0,1)$，那么在上述四条规则下的阈值选取如下例所示。

【例 6-13】计算上述各阈值。

例程 6-13

```
y=randn(1,1000);
thr1=thselect(y,'rigrsure')
thr1=
    2.7316
thr2=thselect(y,'sqtwolog')
thr2 =
    3.7169
thr3=thselect(y,'heursure')
thr3 =
    3.7169
thr4=thselect(y,'minimaxi')
thr4 =
    2.2163
```

因为信号 y 是一个标准高斯白噪声，所以希望每一种方法都能粗略地将所有系数剔除。对于 Stein 的无偏似然估计（SURE）和极大极小（minimaxi）原理的阈值选择规则，仅保存了约 3%的系数，而另外两种阈值选择规则，将所有的系数都变成了零。

同样地，对噪声进行小波分解时，也会产生高频系数。所以一个信号的高频系数向量是有用信号和噪声信号的高频系数的叠加。由于 SURE 和 minimaxi 阈值选取规则较为保守（仅将部分系数置为零），因此在信号的高频信息有很少一部分在噪声范围内时，这两种阈值非常有用，可以将弱小的信号提取出来，另外两种阈值选取规则，在去除噪声时更为有效，但是也可能将有用信号的高频部分当做噪声信号去除掉。

在实际的工程应用中，大多数信号可能包含着许多尖峰或突变，而且噪声信号也并不是平稳的白噪声。对这种信号进行消噪处理时，由于传统的傅里叶变换完全是在频率域中对信号进行分析，它不能给出信号在某个时间点上的变化情况，因此分辨不出信号在时间轴上的任何一个突变。但是小波分析能同时在时频域内对信号进行分析，所以它能有效地区分信号中的突变部分和噪声，从而实现对非平稳信号的消噪。下面我们再来分析一个实例，考察小波分析对非平稳信号的消噪作用。

【例 6-14】请利用小波分析对一个染噪的矩形波信号进行消噪处理。

例程 6-14

```
%设置信噪比和随机值
snr=4;
```

```
init=2055615866;
================================
%产生原始信号 sref 和被高斯白噪声污染的信号 s
[sref,s]=wnoise(1,11,snr,init);
================================
%用 sym8 小波对信号 s 进行 3 层分解并对细节系数
%选用 SURE 阈值模式和尺度噪声
xd=wden(s,'heursure','s','one',3,'sym8');
================================
%对上述信号进行图示
subplot(3,1,1);
plot(sref);
title('参考信号');
subplot(3,1,2);
plot(s);
title('染噪信号');
subplot(3,1,3);
plot(xd);
title('消噪信号');
```

结果如图 6-23 所示。

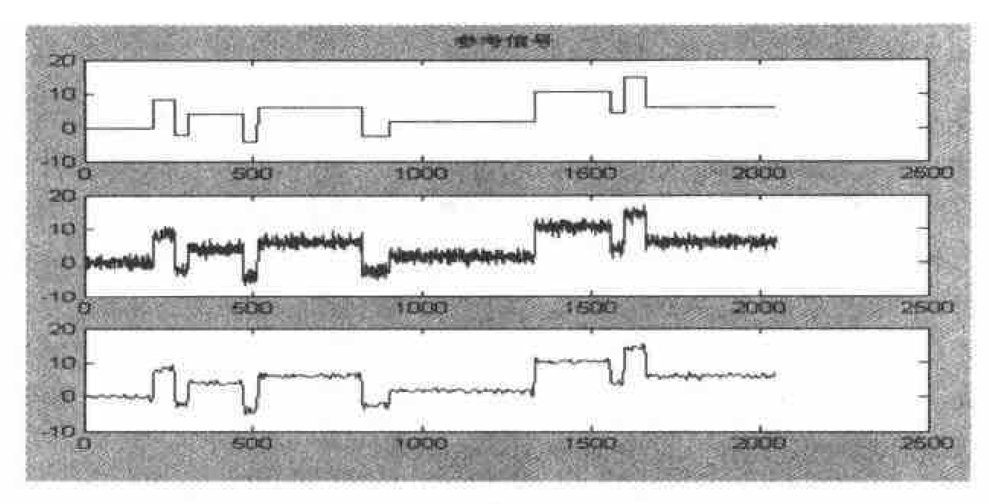


图 6-23　运行结果

4．应用小波包分析进行信号消噪处理

在小波包分析中，其信号消噪的算法思想和在小波分析中的基本相同，所不同的就是小波包提供了一种更为复杂、也更为灵活的分析手段。因为小波包分析对上一层的低频部分和高频部分同时进行分解，具有更加精确的局部分析能力。

对信号进行小波包分解时，可以采用多种小波包基。通常根据分析信号的要求，从中选择最好的一种小波包基，即最优基。最优基的选择标准是熵标准。在 MATLAB 的小波工具箱中可通过 besttree 函数进行最优基的选择，即就是计算最佳树。

应用小波包分析对信号进行消噪处理是它的一个最基本的功能。一般地，按照如下步骤进行：

（1）信号的小波包分解。选择一个小波并确定所需分解的层次，然后对信号进行小波包分解。

（2）确定最优小波包基。对于一个给定的熵标准，计算最佳树。这一步不是必需的步骤，可根据不同的目的进行有选择性的使用。

（3）小波包分解系数的阈值量化。对于每一个小波包分解系数，选择一个恰当的阈值并对系数进行阈值量化。

（4）信号的小波包重构。根据最低层的小波包分解系数和经过量化处理系数，进行小波包重构。

在上述的各步中，最关键的是如何选取阈值和如何进行阈值量化，在一定程度上，它直接关系到对信号进行消噪处理的质量。

【例 6-15】 请利用小波包分析对一个给定的染噪信号进行消噪处理。

例程 6-15

```
%装载原始信号并图示之
load noismima;
s=noismima(1:1000);
subplot(2,2,1);
plot(s);
title('原始信号');
==============================
%采用默认阈值、用 wdencmp 函数进行消噪处理
[thr,sorh,keepapp,crit]=ddencmp('den','wp',s);
==============================
%用全局阈值选项进行消噪处理
[c,treed,perf0,perfl2]=...
    wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,2,3);
plot(c);
title('默认阈值消噪信号');
==============================
%根据前面的消噪效果，调节阈值大小进行消噪
thr=thr+15;
[c1,treed,perf0,perfl2]=...
    wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,2,4);
plot(c1);
title('调节阈值后的消噪信号');
```

结果如图 6-24 所示。

在信号分析中许多情况下都需要提取弱信号，这一点在 FT 分析中是办不到的。例如，在机器故障检测与诊断中，当机器发生故障时，由于机器各零部件的结构不同，致使振动信号所包含不同零部件的故障频率分布在不同的频段范围内。如机器中潜伏着某一零部件

的早期微弱缺陷时，该缺陷信息被其他零部件的运行振动信号和随机噪声所淹没。为了有效地提取弱故障信息及提取某一弱信号，实现早期诊断，可以用小波分析理论，对信号进行小波和小波包分解，把信号分解为各个频段的信号，再根据诊断的目的选取包含所需零部件故障信息的频段序列，进行深层信息处理以查到机器的故障源。

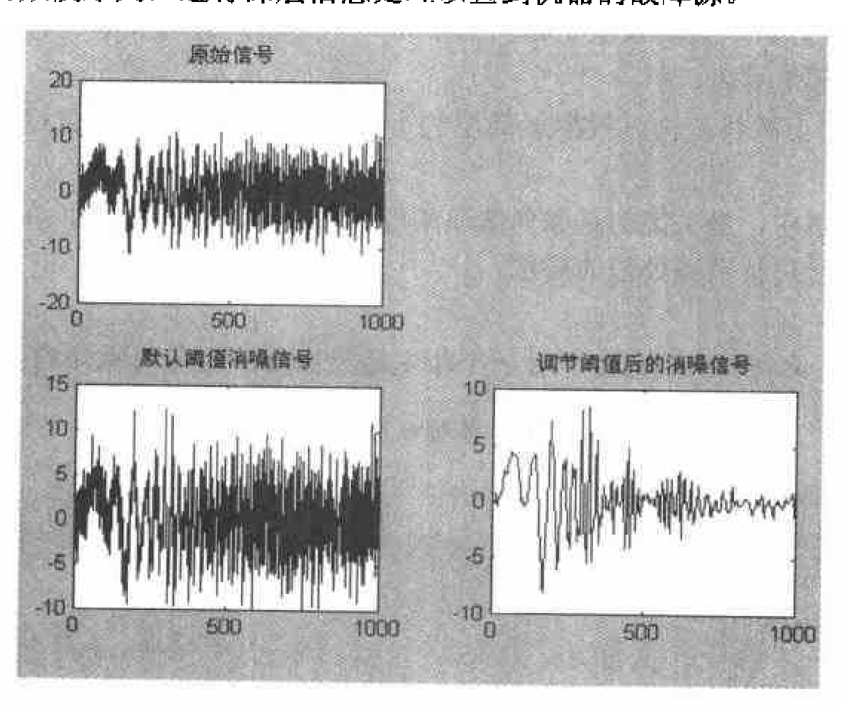


图 6-24　运行结果

要想知道某一频段内信号的频率成分，如果开始采样后给出的信号用尺度函数展开的系数是用小波变换法给出的，那么分解后各频段信号的小波系数本身就是在某一小时间区间内该频段的频率含量。

6.4.7　信号的压缩

应用一维小波分析之所以能对信号进行压缩，是因为一个比较规则的信号是有一个数据量很小的低频系数和几个高频层的系数所组成的。这里对低频系数的选择有一个要求，即需要在一个合适的分解层上选取低频系数。对一维信号进行压缩，可以选用小波分析和小波包分析两种手段进行，它们都可分为以下几个步骤进行：

（1）信号的小波（或小波包）分解；

（2）对高频系数进行阈值量化处理。对第一到第 N 层的高频系数，均可选择不同的阈值，并且用硬阈值进行系数的量化；

（3）对量化后的系数进行小波（或小波包）重构。

信号的压缩与信号的消噪相比，主要差别在第（2）步。一般地，有两种比较有效的信号压缩方法，第一种方法是对信号进行小波尺度的扩展，并且保留绝对值最大的系数。在这种情况下，可以选择使用全局阈值，压缩性能或相对二范数恢复性能，此时仅需要输入一个参数即可。第二种方法根据分解后各层的效果来确定某一层的阈值，且这些阈值可以是互不相同的。

1. 利用小波分析进行信号压缩

下面我们给出一个具体的例子，以便使读者对小波分析的这个重要应用有一个直观的认识。

【例 6-16】 请利用小波分析对给定信号进行压缩处理。

例程 6-16

```
%装载信号
load leleccum;
===========================
%截取信号中的一段[2600，3100]
s=leleccum(2600:3100);
===========================
%用小波 db3 对 s 进行 3 层分解
[c,l]=wavedec(s,3,'db3');
===========================
%选用全局阈值进行信号压缩处理
thr=40;
[sd,csd,lsd,perf0,perfl2]=wdencmp('gbl',c,l,'db3',3,thr,'h',1);
subplot(2,1,1);
plot(s);
title('原始信号');
subplot(2,1,2);
plot(sd);
title('压缩后的信号');
```

结果如图 6-25 所示。

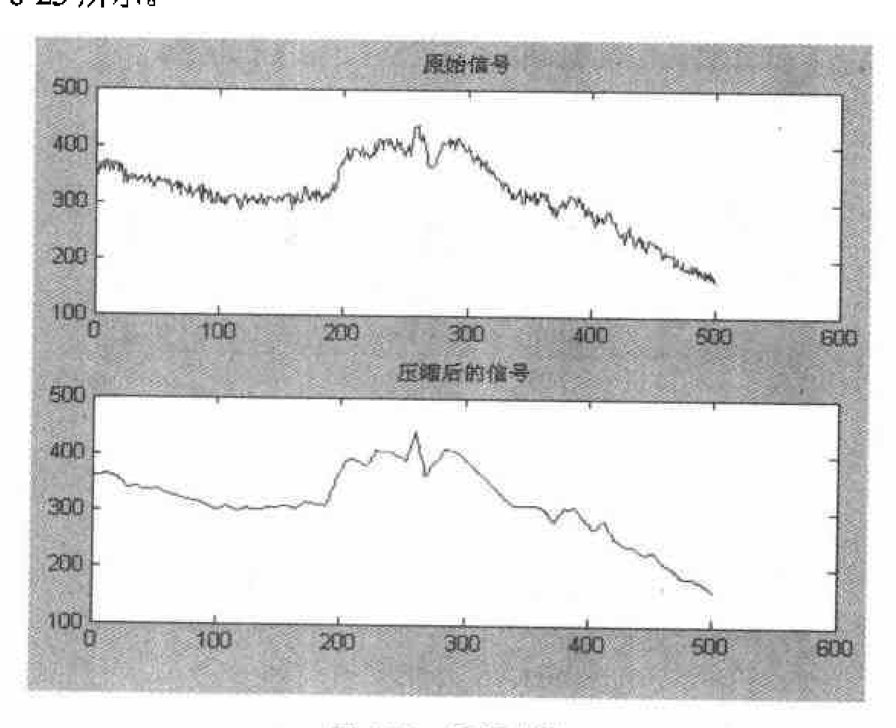


图 6-25　运行结果

2. 利用小波包进行信号的压缩处理

在小波包分析中，其信号压缩的算法思想和在小波分析中的基本相同，所不同的就是小波包提供了一种更为复杂，也更为灵活的分析手段。因为小波包分析对上一层的低频部分和高频部分同时进行分解，具有更加精确的局部分析能力。

在用小波包分析进行信号压缩处理中，最关键的是如何选取阈值和如何进行阈值量化，在一定程度上，这直接关系到对信号进行压缩处理的质量。

【例 6-17】利用小波包分析对一个给定的一维信号进行压缩处理。

例程 6-16

```
%装载源信号
load noisbump;
s=noisbump(1:1000);
subplot(2,1,1);
plot(s);
title('原始信号');
=============================
%采用默认阈值，以小波包函数 wpdencmp 对 s 进行压缩处理
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',s);
[sc,treed,perf0,perfl2]=...
    wpdencmp(s,sorh,3,'db2',crit,thr,keepapp);
subplot(2,1,2);
plot(sc);
title('压缩后的信号');
```

结果如图 6-26 所示。

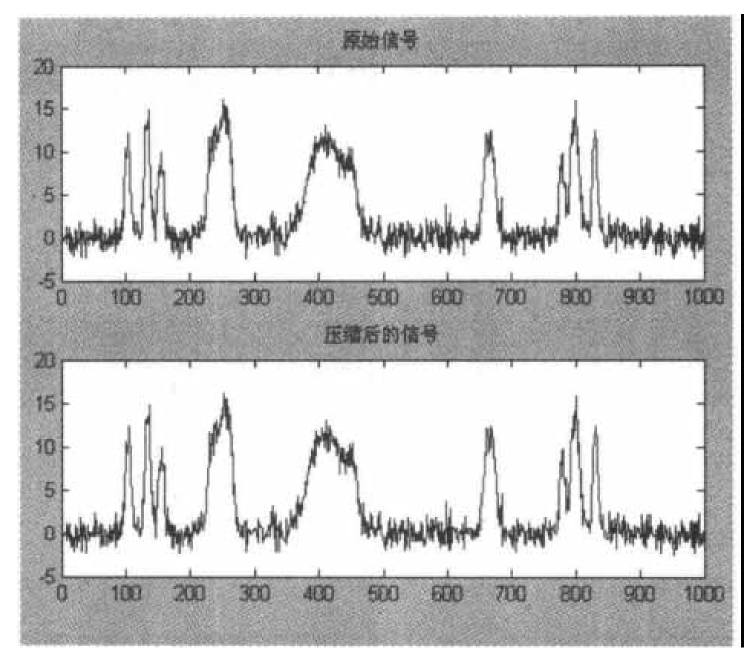


图 6-26 运行结果

至此，我们已经全面地向读者介绍了小波分析在信号处理领域中的应用。有兴趣的读



者可以对以上实例中的小波、分解层次、阈值等进行改变并上机操作，观察所得结果与实例中结果的异同，以达到掌握应用小波分析进行信号处理的目的。

6.5 应用图形界面（GUI）方式进行信号处理

前面我们介绍了应用命令行形式如何对信号进行处理的知识。在本节，我们介绍一下 MATLAB 小波分析工具箱中的一个具有强大功能，且极其方便的人机交互形式的处理信号方式——GUI 方式（Graphical User Interface）。它可以实现小波分析的绝大多数功能，不需编程，只要通过菜单操作和选择参数就可对一维或二维信号进行处理，使用起来更加简单、直观。下面，我们就上一节中的各种要求对信号进行相关处理。

首先，我们在 MATLAB 命令行窗口键入命令 wavemenu，然后单击【Wavelets 1.D】按钮，就可进行您所需要做的工作。

6.5.1 第一种类型间断点的检测

在这里所用的信号是由一个低频正弦波后跟随一个中频正弦波组成，所用分析小波为 db5，且分解到第 5 层，如图 6-27 所示。

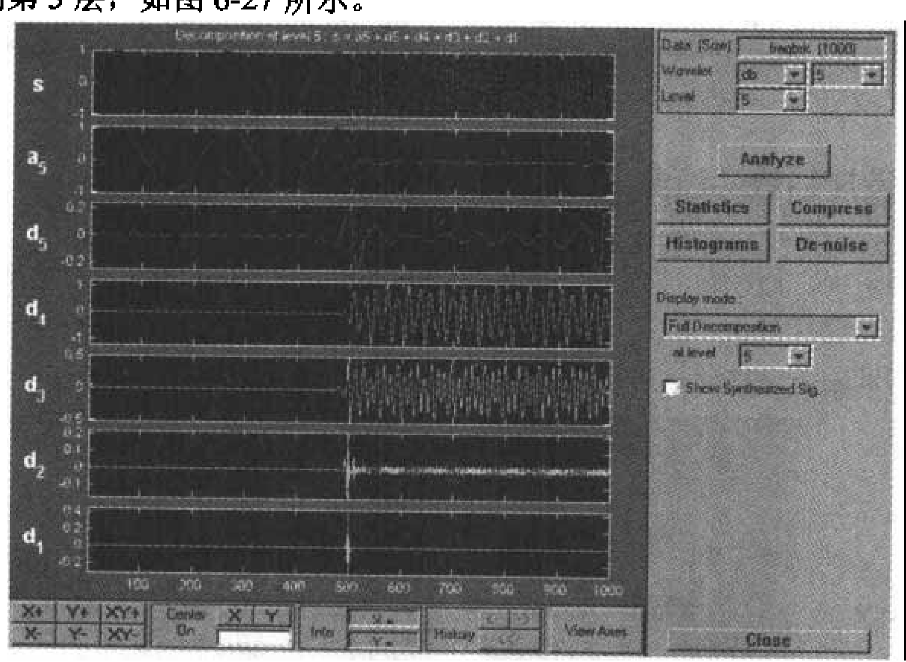


图 6-27 第一种类型间断点的检测

由于该信号中的突变发生在较高频率部分，所以在分解后的第一、二层细节中非常清晰地显示了该信号中所包含的第一种类型间断点。

6.5.2 第二种类型间断点的检测

在图 6-28 中的两幅图中，分别给出了应用小波分析进行第二种类型间断点检测时所遇到的不同阶次导数不连续的情况（其中，左图是一阶导数不连续的情况，右图是二阶导

数不连续的情况)。

在这种处理要求下，对所用分析小波的正则性有所要求。一般地，在信号的第 j 阶导数上有突变，则所选用的分析小波至少应具有 j 个消失矩。

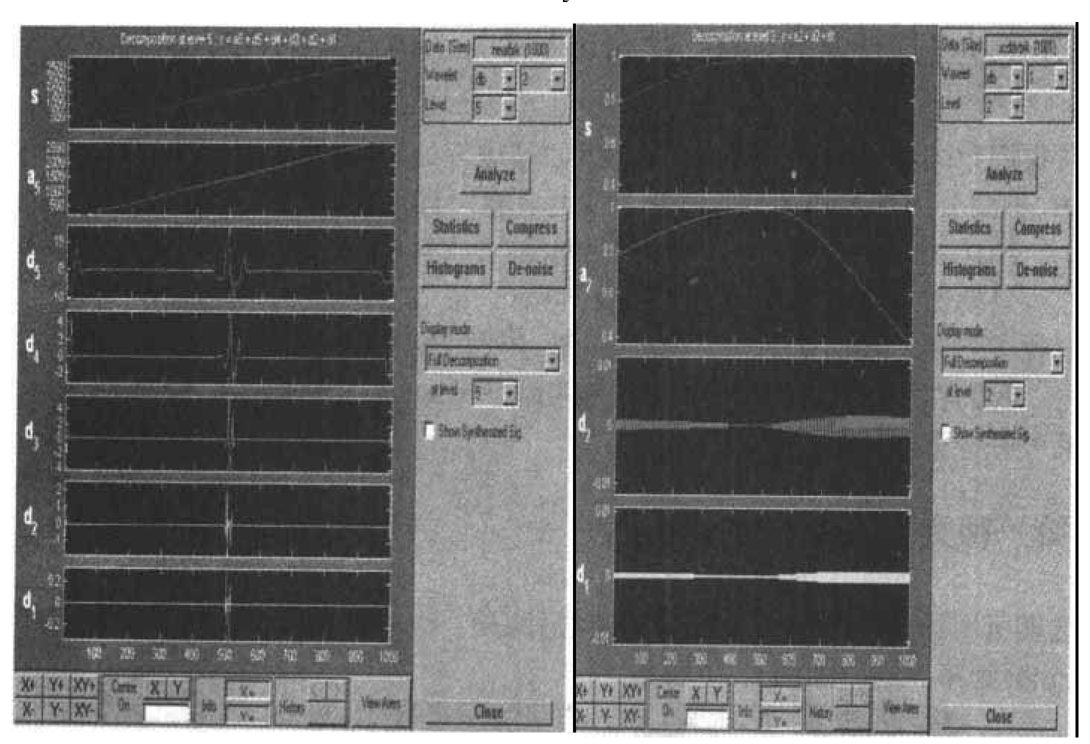


图 6-28　第二种类型间断点的检测

6.5.3　信号发展趋势的检测

在这个例子中，原始信号被高频噪声信号所淹没，关于原始信号的发展趋势我们是无法分辨的。通过选择分析小波和分解层次，我们可以看出随着分解层次的加深，分解系数的近似部分越来越清楚地显示了原始信号的发展趋势，如图 6-29 所示。

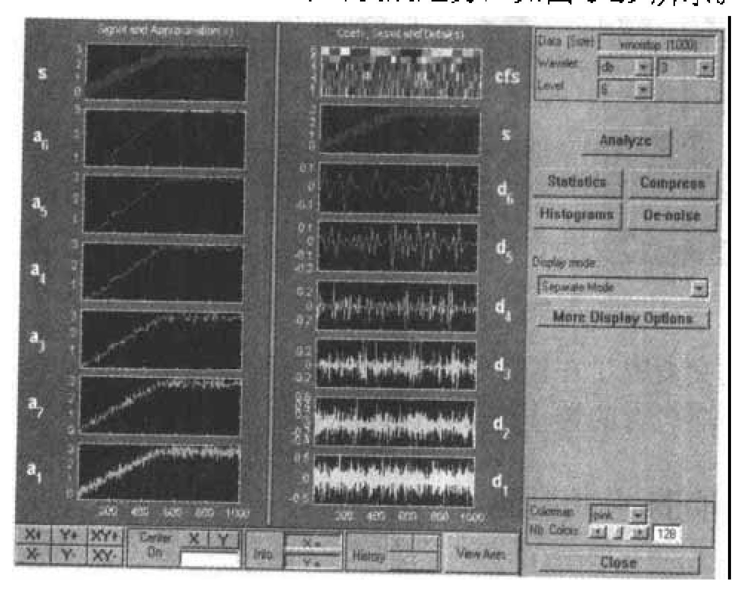


图 6-29　信号发展趋势的检测



6.5.4 信号自相似性的检测

图 6-30 显示了信号自相似性检测的结果。在这里需说明的是，所用分析小波为一维连续小波，而在小波系数的图形中，重复模式正表明了该信号在许多尺度下看起来相似的特征。直观上，检测信号自相似性的小波分解算法是计算信号和小波之间的自相似指数。如果这个指数大，则它们的相似程度高。反之亦然。这里的自相似指数就是小波系数。如果一个信号在不同的尺度上均相似于它本身，那么自相似指数或者说小波系数在对应的尺度上也是相似的。

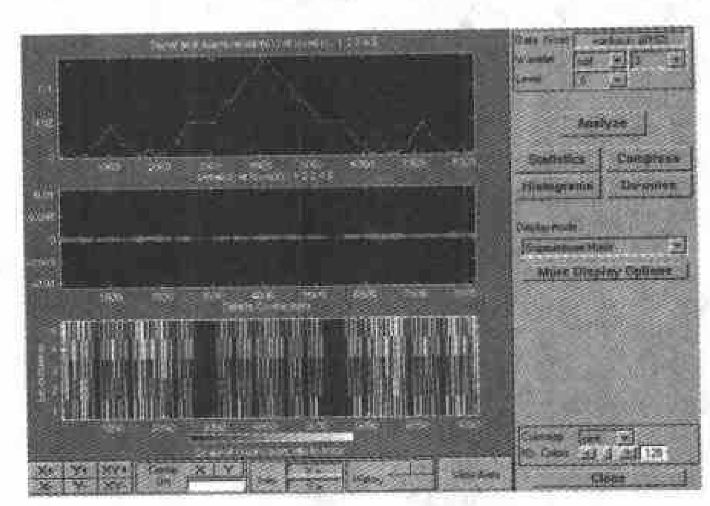


图 6-30 信号自相似性的检测

许多学者所做的工作表明，小波分解用于信号自相似性检测对信号和图像的分形特性研究提供了重要的思路。

6.5.5 信号识别

图 6-31 显示了小波分析用来识别位于不同频率区间的信号的功能。从小波分解后的各层细节和近似中，我们能够分辨出原始信号中位于不同频率区间的各个分信号。

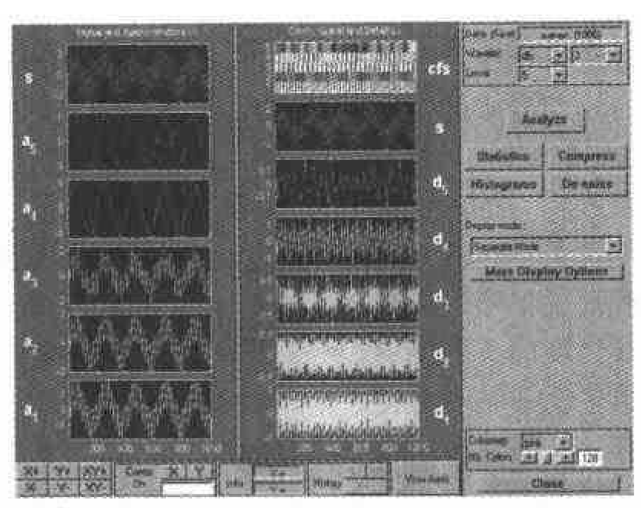


图 6-31 信号识别

6.5.6 信号抑制

图 6-32 所示的例子目的是显示一个多项式经过适当次数的小波分解（所用分析小波的消失矩数大于多项式的次数即可）后可得到空细节的特性。这里所用信号为一个带有微弱白噪声的二次多项式。

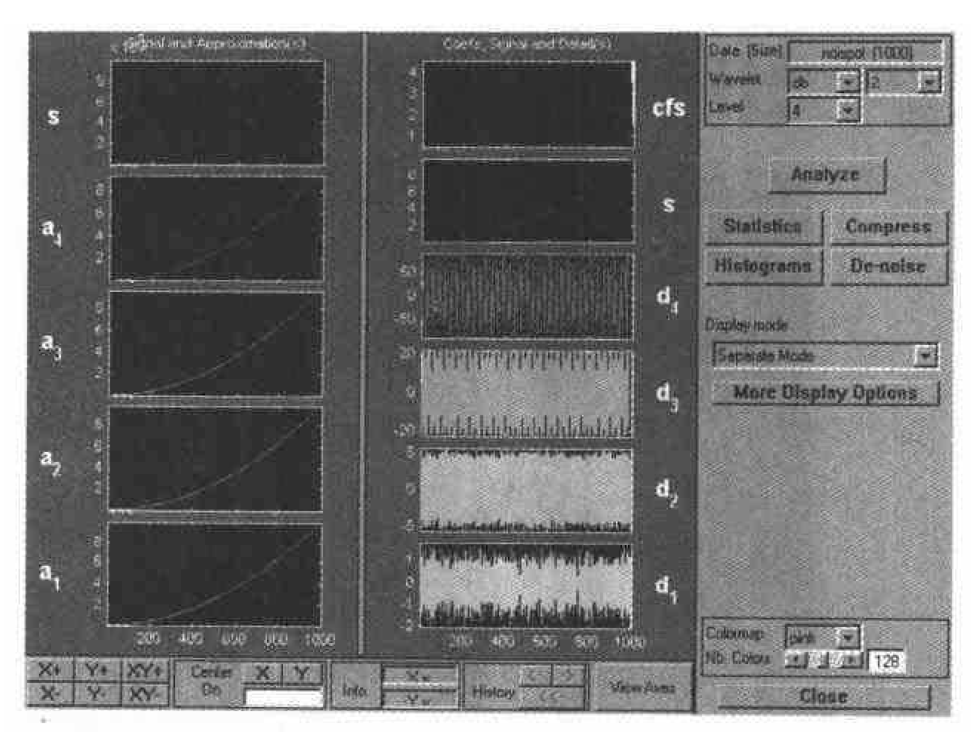


图 6-32 信号抑制

6.5.7 信号消噪

这个例子显示了利用小波分析如何对一个信号进行消噪。在这个例子中也给出了一维小波图形界面自动设置阈值的功能。首先看图 6-33，利用小波 sym4 对信号 Noisy Doppler 进行 5 层分解，仔细观察分解后的各层近似和细节图形，可以看出随着分解层次的加深，包含在近似中的噪声信息越来越少，而在第五层中，几乎所有的噪声信号被除掉了。但是，同时不可避免地丢掉了有用信号中的一些高频信息。

用鼠标单击【De-noise】按钮，就会出现如图 6-34 所示的窗口，自动地显示每一层细节中所设置的阈值，并且在窗口的右上角可以发现消噪后的信号和原始信号被绘在同一帧图形中，消噪后的信号用黄色曲线表示，而原始信号用红色曲线表示，在本书中，颜色较亮的曲线表示的就是消噪后的信号。同样地，在信号开始的时间段内的高频信息也丢失了一部分，因为此时的曲线呈现出了平坦的趋势，尽管如此，和图 6-33 中的近似 a4 和 a5 相比，丢失的信息则少得多。

对于信号的消噪处理，小波包分析比小波分析的精度高得多。有兴趣的读者可以使用 GUI 中的一维小波包工具对信号进行消噪处理。这里推荐一种演示模式：使用 Wavelet Packet 1.D 工具，在菜单中选择菜单命令【File】→【Demo Analysis】→【noisdopp】。

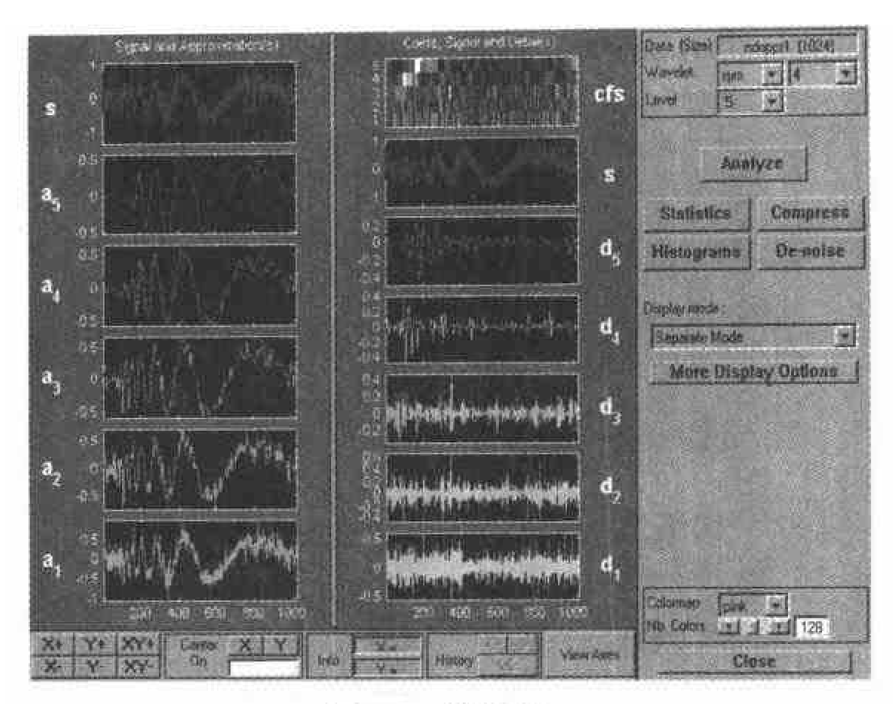


图 6-33　信号消噪 1

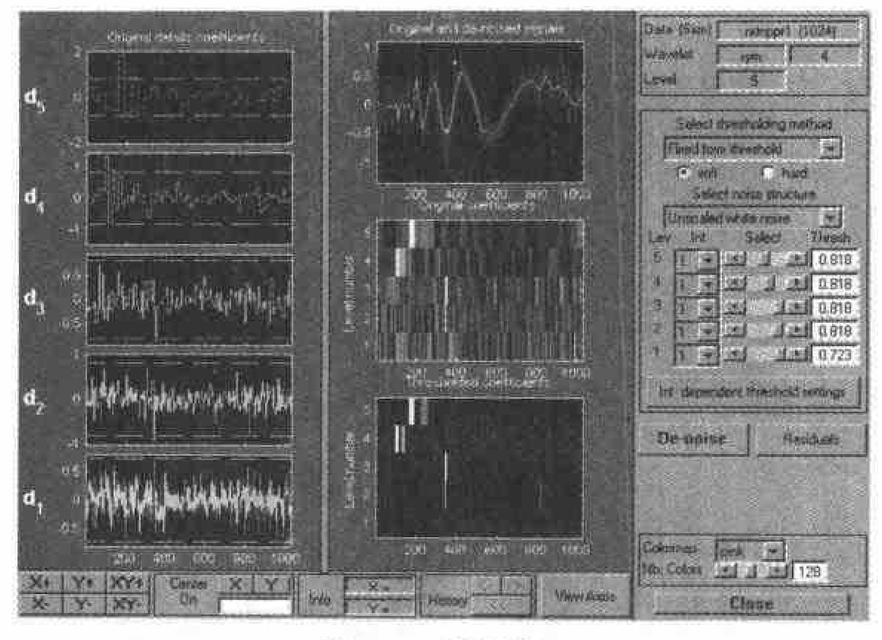


图 6-34　信号消噪 2

6.5.8　信号压缩

图 6-35 和图 6-36 给出了应用 GUI 进行信号压缩的结果。当我们选择好分析小波并对所给信号进行分解处理时，同时也就确定了各层细节的阈值，如图 6-35 所示。然后用鼠标单击图 6-35 所示的窗口中的【Compress】按钮，就会出现图 6-36 所示的窗口。在该窗口的右上角会发现压缩回的信号和原始信号绘在了同一帧图形中，其中黄色曲线表示的是压缩后信号，而红色曲线表示的是原始信号，在本书中，压缩后信号用颜色较亮的曲线表

示。同时，这两个窗口中均显示了所用阈值和压缩百分比。

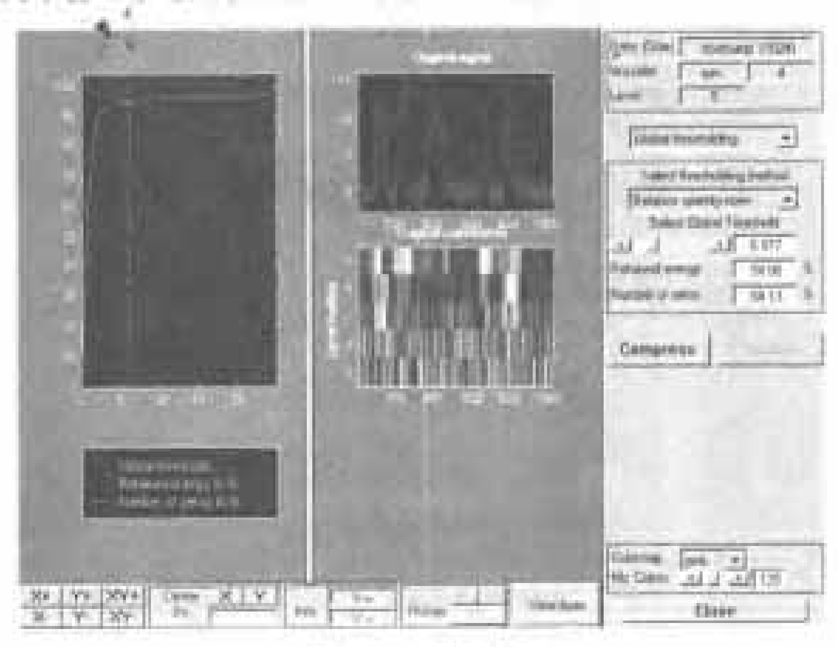


图 6-35　信号压缩 1

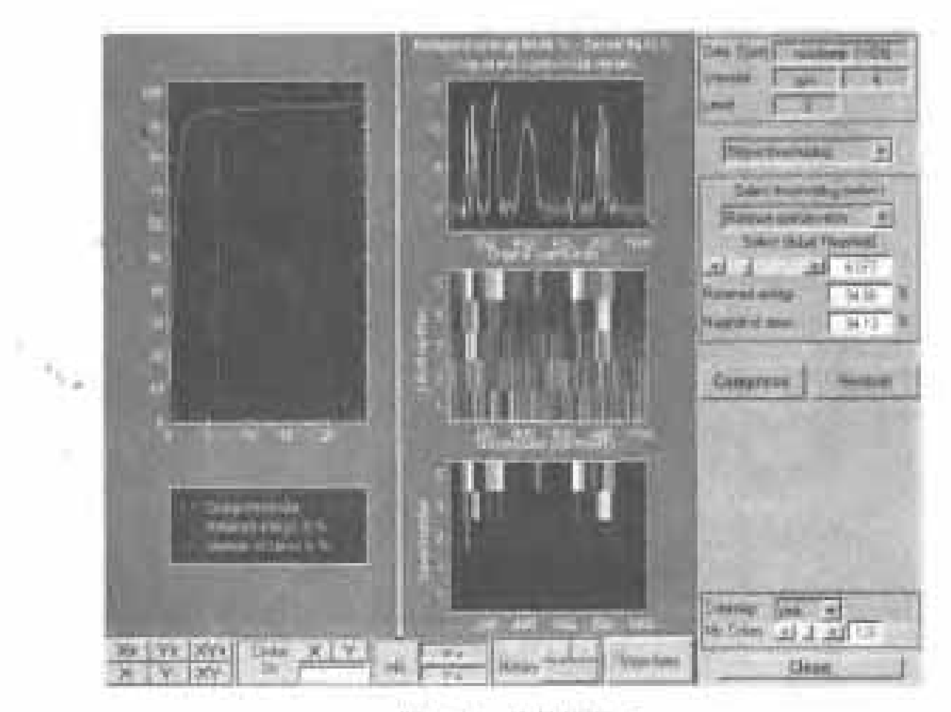


图 6-36　信号压缩 2

第 7 章　小波分析用于图像处理

图像处理是小波分析应用的重要领域，近年来小波分析已被证明是进行图像处理强有力的工具之一，因此下面将结合 MATLAB 6.5 语言介绍小波分析在图像处理这一领域的应用。

本章主要内容：

- 计算机数字图像文件
- 图像编码方式和处理概述
- 二维小波分析用于图像处理
- 应用 GUI 进行图像处理

7.1　计算机数字图像文件

7.1.1　数字图像文件格式

1．计算机数字图像文件常用格式

- BMP（Windows Bitmap）格式

这种格式包括有 1、4、8、24 位非压缩图像，8 位 RLE（Run.length Encoded）图像格式。文件内容包括有：文件头（一个 BITMAP FILEHEADER 数据结构）、位图信息数据块（位图信息头 BITMAP INFOHEADER 和一个颜色表）和图像数据。

- PCX（Windows Paintbrush）格式

它可处理 1、4、8、16、24 位等图像数据。文件内容包括：文件头（128 字节）、图像数据和扩展调色板数据。

- JPEG（Joint Photographic Experts Group）格式

这是一种称为联合图像专家组的图像压缩格式。

- TIFF（Tagged Image File Format）格式

能够处理 1、4、8、24 位非压缩图像，1、4、8、24 位 packbit 压缩图像，1 位 CCITT 压缩图像等。其文件内容包括有：文件头、参数指针表与参数域、参数数据表和图像数据四部分。

- TGA 格式

这种格式可处理 1、4、8、16、24 位非压缩图像和行程编码（RLE）图像。文件包由 5 个固定长度字段和 3 个可变长度字段组成。

- XDW（X Windows Dump）格式

1 位和 8 位 Zpixmaps 和 XYBitmaps，1 位 XYPixmaps。

- HDF(Hierarchical Data Format)格式

该格式包含有 8 位、24 位光栅图像数据集。

2. MATLAB 支持的基本图像类型

● 索引图像

索引图像包括图像矩阵与颜色图数组，其中颜色图是按图像中颜色值进行排序后的数组。对于每个像素，图像矩阵包含一个值，这个值就是颜色图数组中的索引。颜色图为 m×3 的双精度值矩阵，各行分别指定红、绿、蓝（R、G、B）单色值，且 R、G、B 均为值域[0，1]上的实数值。

图像矩阵与颜色图的关系依赖于图像矩阵是双精度类型还是 unit8（无符号 8 位整数）类型。如果图像矩阵为双精度类型，则第一点的值对应于颜色图的第一行，第二点的值对应于颜色图的第二行，依次类推，各个点的值都对应于相应颜色图的各个行。如果图像矩阵是 unit8，且有一个偏移量，即第 0 点的值对应于颜色图的第一行，第 1 点的值对应于颜色图的第二行，依次类推，则 unit8 类型常用于图形文件格式，且它支持 256 色。

● RGB 图像

与索引图像一样，RGB 图像也是分别用红、绿、蓝三个亮度值为一组，代表每个像素的颜色。与索引图像不同的是，这些亮度值直接存在图像数组中，而不是存放在颜色图中。图像数组为 $m \times n \times 3$，m，n 表示图像像素的行列数。

● 二进制图像

在二进制图像中，每个点为两个离散值中的一个，这两个值代表开或关。二进制图像被保存在一个二维的由 0（关）和 1（开）组成的矩阵中。从另一个角度讲，二进制图像可以看成为一个仅包括黑与白的特殊灰度图像，也可看做仅有两种颜色的索引图像。

二进制图像可以保存为双精度或 unit8 类型的数组，显然使用 unit8 类型更节省空间。在图像处理工具箱中，任何一个返回二进制图像的函数都是以 unit8 类型逻辑数组来返回的。

● 灰度图像

在 MATLAB 中，灰度图像是保存在一个矩阵中的，矩阵中的每一个元素代表一个像素点。矩阵可以是双精度类型，其值域为[0，1]；矩阵也可以是 unit8 类型，其数据范围为[0，255]。矩阵的每一个元素代表不同的亮度或灰度级，其中，亮度为 0，表示黑色，亮度为 1（或者 unit8 类型的 255），则代表白色。

7.1.2 MATLAB 小波工具箱与索引图像

在 MATLAB 中，基本的数据结构是矩阵，其元素是一组有序的实数或复数，这样可以方便地表示图像。但是，在小波工具箱中，仅仅支持有序实数所表示的图像。像素是图像显示的基本单元，表示图像矩阵中的一个元素；而在计算机的显示器上，像素代表了一个点。因此，通过一般矩阵下标从图像矩阵中可以很容易地选取一个像素。例如，输入命令 I（2，15），即可返回图像中第 2 行、第 15 列的像素值。默认情况下，MATLAB 会按比例显示图像，使其充满整个坐标轴，因此，图像的一个像素可能使用几个屏幕上的像素点。

一个典型的彩色图像需要两个矩阵来表示，即颜色图和图像矩阵。颜色图是一组用来

表示图像中颜色的有序数值。对每一个图像像素来说，图像矩阵包含了颜色图的相应索引。在此需注意，图像矩阵的元素是浮点型整数，而 MATLAB 将其作为双精度型数值存储。对一个包含 n 个颜色的图像，其颜色图矩阵是 $n\times3$ 的，且颜色图矩阵的每一行是 1×3 的向量，即 ***color***=[R G B]，其分量分别代表红、绿、蓝三种颜色的状态，R、G 和 B 是从 0 到 1 变化的实数。当显示一个图像时，MATLAB 会将这些值转换成其所对应的颜色的显示状态，具体的转换过程可参照图 7-1。MATLAB 显示索引图像时，使用图像矩阵中的值在颜色图中查找所需的颜色，如图 7-1 所示。如果图像矩阵中位置（86，198）上的值是 18，那么，像素（86，198）的颜色就是颜色图中第 18 行所代表的颜色。

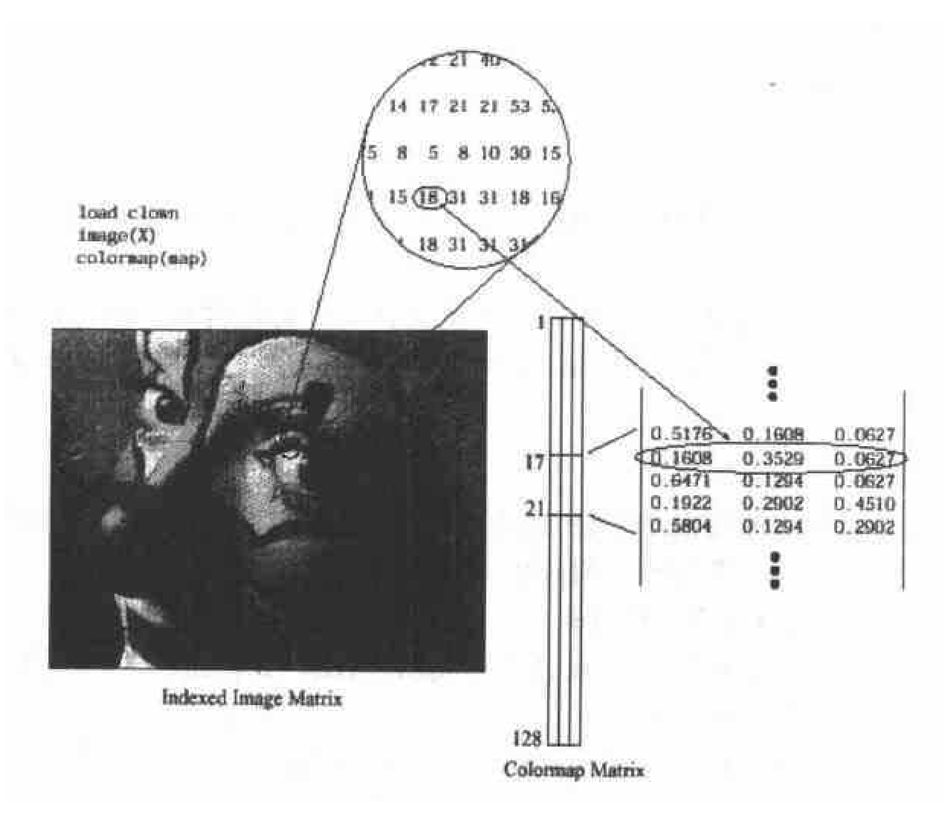


图 7-1 索引图像的表示



在 MATLAB 中，具有 n 种色彩的索引图像的索引值是从 1 到 n 变化的。而其他语言中，该值是从 0 到 n.1 变化的。

1．索引图像的小波分解

小波工具箱仅支持具有线性单调变化的颜色图的索引图像。这些图像可以看做是按比例变化的颜色状态图，其矩阵元素仅仅包含从 1 到 n 的整数，而 n 表示图像中的离散色彩数。如果在图像矩阵中没有颜色图，那么图形用户界面（GUI）工具利用单调变化的颜色图（具有 max(max(X)).min(min(X))+1 种色彩）来显示图像和处理结果。由于图像颜色图仅用做显示的目的，因此，为了从小波分解中得到正确的结果，一些索引图像可能需要进行预处理。一般地，彩色索引图像并不具有线性单调变化的颜色图，因此，在进行小波分解之前，必须将该彩色索引图像转换成灰度索引图像。

索引图像经过小波分解后，所得到的小波系数、近似和细节并不是索引图像矩阵。因



此为了用合适的方式显示图像，应遵循下述规则：

- 使用颜色图显示重构近似；
- 按照 GUI 中的矩阵重组方式，使用颜色图显示系数和重构细节。

2．其他类型图像的小波分解

小波工具箱也允许处理一些其他类型的图像。如果使用 imread 函数，则小波工具箱提供了下面几种图像分析工具，可以对那些来自非 MAT 文件（例如 PCX 文件等）的索引图像进行分解。

这些工具有：

- 二维离散小波分析
- 二维小波包分析
- 二维固定小波分析
- 二维扩展工具

如果读者有兴趣的话，可以在 MATLAB 命令窗口键入：help imread，以查看更多的信息。

在对其他类型的图像进行小波分析时，如果文件中没有包含索引图像，那么，装载操作就会失败。在这种情况下，可通过 imfinfo 函数查看保存在文件中的图像类型，以避免出错。

3．图像类型转换

为了使读者能够充分理解MATLAB小波工具箱处理图像的强大功能，这里对MATLAB图像处理工具箱中有关图像类型转换的函数做一简要介绍。

（1）灰度图像与索引图像的相互转换函数：gray2ind()，ind2gray

- [X，MAP]=gray2ind(I，n)，其作用是将灰度图像 I 转换成具有颜色图 gray(n)的索引图 X，n 的默认值为 64。
- I=ind2gray(X，MAP)，其作用是将具有颜色图 MAP 的图像 X 转换成灰度图像，X 可以是 unit8 或双精度类型，I 为双精度类型。

（2）RGB 图像转换为灰度图像函数：rgb2gray()

I=rgb2gray(RGB)，其作用是将真彩图像 RGB 转换为灰度级亮度图像 *I*。

（3）RGB 图像与索引图像相互转换函数：rgb2ind()，ind2rgb()

- 函数 rgb2ind() 将 RGB 图像转换为索引图像，采用的方法有：直接转换、均匀量化、最小方差量化和颜色图近似。除直接转换外，其他方法在不指定选项'nodither'时自动进行图像抖动。
- RGB=ind2rgb(X，MAP)将矩阵 X（unit8 或双精度格式）及其相应的颜色图 MAP 转换成 RGB 真彩色格式（m×n×3 双精度数组）图像。

（4）将图像转换为二进制图像函数：im2bw()

使用该函数可将索引、灰度和 RGB 图像转换成为二进制（黑白）图像。在转换过程中，如果输入图像不是灰度级图像，首先将其转换为灰度级图像，然后，通过阈值化将灰度级图像转换成二进制图像。输出图像在输入图像所有亮度小于给定值（level）像素点处均为 0，在其他地方均为 1。

（5）从灰度图像产生索引图像函数：grayslice()

- X=grayslice(I，n)，其作用是使用阈值 1/n,2/n,⋯,(n-1)/n 来阈值化灰度图像 I，产生一个索引图像 X。
- X=grayslice(I，V)，其作用是 V 为一个元素值在[0，1]上的向量，使用 V 中的值作为门限 I 的阈值，产生一个索引图像 X。

4．MATLAB 图像类型转换

首先介绍 MATLAB 中支持的几种图像类型转换。

（1）先调入并显示一个 RGB 图像：

```
load myimage;
%判断该图像是否为 RGB 图像
isrgb(X)；%X 是图像矩阵
%绘出该图像
figure(1)
image(X);
```

（2）将 RGB 图像转换为索引图像：

```
[Y，MAP]=rgb2ind(X,256);
figure(2)
image(Y);
colormap(MAP);
colorbar;
```

（3）RGB 图像转换为灰度级图像：

```
I=rgb2gray(X);
imshow(I);
colorbar;
isgray(I)；%判断 I 是否为灰度级图像
```

下面给出两个例子说明图像类型的转换。

【例 7-1】将彩色索引图像转换为灰度级索引图像。

```
load xpmndrll
whos    %查看图像矩阵和颜色图名称
  Name        Size              Bytes   Class

  X2        192×200          307200   double array
  map        64×3              1536   double array

image(X2)
title('原始彩色索引图像')
colormap(map);
colorbar
```

如图 7-2 所示，图像右边的色彩条不是平滑的，也不是从暗到亮单调变化的。这种索

引图像不适合于直接进行小波分解，需要对该图像进行预处理。首先我们把彩色索引图像分解为 RGB 分量：

```
%分解彩色索引图像为 RGB 分量
R=map(X2,1);
R=reshape(R,size(X2));
G=map(X2,2);
G=reshape(G,size(X2));
B=map(X2,3);
B=reshape(G,size(X2));
```

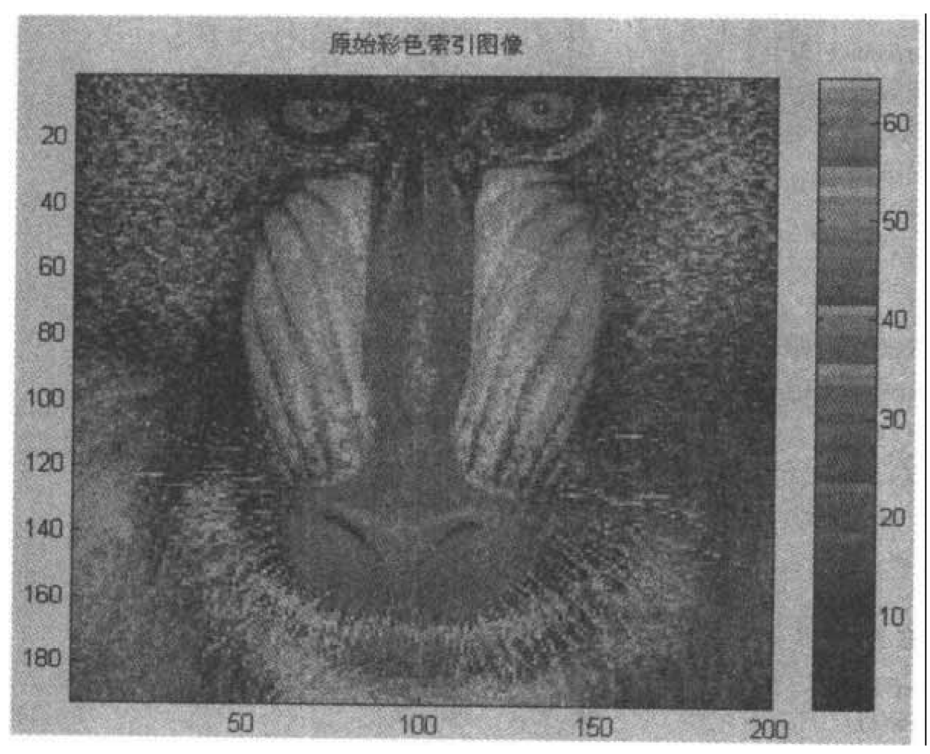


图 7-2　运行结果

接下来，我们使用标准的权值对这三种颜色分量进行加权处理，将 RGB 矩阵转换成为灰度级状态图像。命令行为：

```
Xrgb=0.2990*R+0.5870*G+0.1140*B;
```

然后，我们将灰度级状态图像转换为灰度级索引图像，该索引图像具有 64 个不同的索引值，同时，建立这 64 个灰度索引的颜色图。命令行如下：

```
n=64;
X=round(Xrgb*(n.1))+1;
map2=gray(n);
figure(2);
image(X);
title('处理后的灰度级索引图像');
colormap(map2);
colorbar
```

处理后的结果如图 7-3 所示，该灰度级索引图像的色彩条是由暗到亮光滑渐变的。这种图像可以进行小波分解处理。最后，我们将转换后的索引图像保存起来，以便于小波工



具箱中的图形用户界面（GUI）使用。

```
baboon=X;
map=map2;
save baboon baboon map
```

===================================

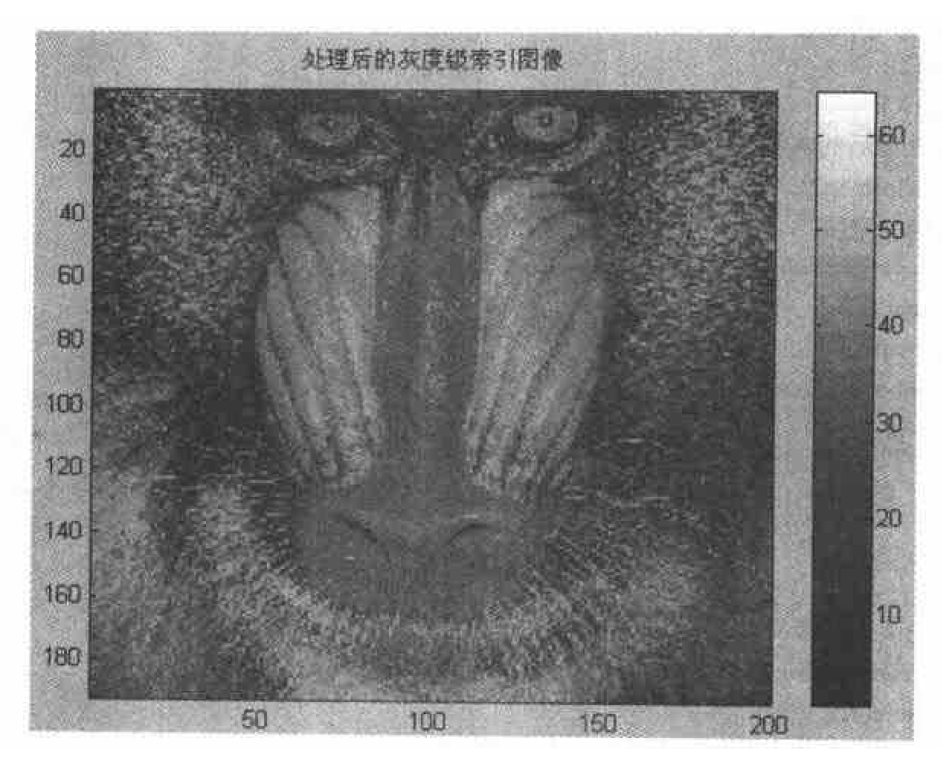


图 7-3　灰度级索引图像

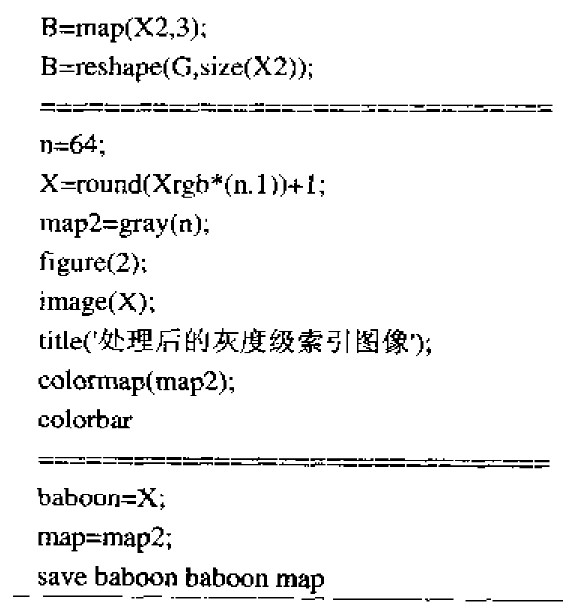
本例的程序清单如例程 7-1。

例程 7-1

```
load xpmndrll
whos %查看图像矩阵和颜色图名称
  Name       Size           Bytes   Class

  X2       192×200         307200   double array
  map       64×3             1536   double array

image(X2)
title('原始彩色索引图像')
colormap(map);
colorbar
===================================
%分解彩色索引图像为 RGB 分量
R=map(X2,1);
R=reshape(R,size(X2));
G=map(X2,2);
G=reshape(G,size(X2));
```



```
B=map(X2,3);
B=reshape(G,size(X2));
================================
n=64;
X=round(Xrgb*(n.1))+1;
map2=gray(n);
figure(2);
image(X);
title('处理后的灰度级索引图像');
colormap(map2);
colorbar
================================
baboon=X;
map=map2;
save baboon baboon map
```

【例 7-2】将 TIF 文件格式的 RGB 图像转换成为索引图像。假设有一个大小为 $S1 \times S2$ 的 RGB 图像（未压缩）包含在文件 myimage.tif 中，请利用 MATLAB 命令转换该图像。

例程 7-2

```
%首先读入图像文件
A=imread('myimage.tif');%A 是一个 unit8 类型的 S1×S2×3 数组
A=double(A);
Xrgb=0.2990*A(:,:,1)+0.5870*A(:,:,2)+0.1140*A(:,:,3);
NbColors=255;
X=wcodemat(Xrgb,NbColors);
map=pink(NbColors);
image(X);
colormap(map);
colorbar;
```

这里仅以 TIF 格式的文件作为例子，对于其他格式，如 BMP 或 JPEG 等格式的文件，该程序同样可以对其进行处理。有兴趣的读者可自行演示上述程序的功能。

7.2 图像编码方式和处理概述

7.2.1 图像编码方式

对于图像来说，由于传输和保存对快速性的要求，因此必须对具有大数据量的原始图像进行压缩处理，并且在传输后还需对图像进行恢复。因此在图像的压缩、传输、保存等过程中，必须要保持图像的特征不发生改变，以便于图像的分类和识别。众所周知，图像数据往往存在各种信息的冗余，如空间冗余、信息熵冗余、视觉冗余和结构冗余等。所谓压缩，就是想法去掉各种冗余，保留真正有用的信息。通常把图像进行压缩的过程称为编

码，恢复图像的过程称为解码。

根据解码后的数据与原始数据是否完全一致来分类，通常将图像压缩方法划分为无失真编码与有失真编码两大类。

1. 有失真编码

有失真编码方法所还原的图像与原始图像相比，存在一定的误差，但视觉效果是可以接受的。根据有失真编码的原理进行分类，可分为预测编码、量化编码、变换编码、信息熵编码、结构编码、基于知识的编码和分频带编码等。

- 预测编码是一种针对统计冗余进行压缩的方法。对于统计冗余来说，它反映图像内相邻两个像素之间的相关性比较，因而一个像素可以由与它相邻的并且已被编码的像素来进行预测估计。当然，这种预测是根据一定的模型进行的。从理论上讲，只要模型选取得足够好，则只需保存或传输起始像素与模型参数就可以代替一幅图像。但是，在实际应用中，预测不会总是正确的和精确的，此时的做法是再将预测误差保存或传输。预测编码方法基本上是针对输入的数据是一个平稳过程而设计的。当输入的数据不是平稳过程时，可采用自适应预测编码。
- 量化编码与向量量化编码的本质都是对统计冗余进行压缩，向量量化是量化的最具代表性的方法。在图像的压缩中，可将图像看做一串数据，将这串数据截成相等的 M 段（即分成 M 个向量），再把这 M 个向量分成 N 组，在每一组中挑选一个数据向量，作为该组的代表。所谓压缩，就是图像上的数据向量，如果属于某个组，则这个数据向量就用该组的代表向量代替，这时的编码就是在编码书的相应位置上记下这个代表向量的编号。
- 变换编码也是一种针对统计冗余进行压缩的方法。所谓变换编码是将图像时域（空间）信号变换到系数空间（频域）上进行处理的方法。因为由时域映射到频域总是通过某种变换进行的，所以该方法称为变换编码。在空间上具有强相关的信号，反映到频域上是在某些特定的区域中将能量集中在一起，或者是系数矩阵的分布具有某种规律，因此就可利用这些规律分配频域上的量化比特数，从而达到压缩的目的。变换编码具有两个最明显的特点，一是可以得到高的压缩比，二是比预测等其他方法的计算复杂性高。在变换后，由于在频域上信息是按照频谱的能量与频率的分布排列的，只要对频域平面量化器进行合理的（非均匀）比特分配，即高能量区给以高的比特数，低能量区给以低的比特数，就可得到高的压缩能力。常用的变换有：KL 变换、DCT 和 DST 变换、DFT 变换、Haar 变换、Walsh.Hadamard 变换，以及用途广泛的小波变换等。
- 信息熵编码是根据信息熵原理，用较短的码字表示出现概率大的信息，用较长的码字表示出现概率小的信息。最常见的有 Huffman 编码、游程编码和算术编码等。
- 结构编码也称为第二代编码。用这种方法编码时，首先将图像中的边界、轮廓和纹理等结构特征求出来，然后保存这些参数信息。解码时，根据结构和参数信息进行合成，从而恢复出原始图像。
- 基于知识的编码是对于人脸等可用规则描述的图像，利用人们对这些图像的知识形成一个规则库，据此将人脸等图像的变化用一般的参数进行描述，从而用参数

与模型就可以实现人脸等图像的编码与解码。

- 分频带编码是将图像数据变换到频域后，按频率划分频带，然后用不同的量化器进行量化，从而达到最优的组合；或者是分步渐进编码，开始对某一频带的信号进行解码，然后逐渐扩展到所有频带。随着解码数据的增加，解码图像也逐渐清晰起来。此方法对于远地图像模糊查询与检索的应用比较有效。

2．无失真编码

无失真编码方法要求在解码后得到的图像与原始图像严格相同，如改进的 Huffman 编码等。

小波分析用于图像压缩是小波分析应用的重要方面之一。其特点是压缩比高，压缩速度快，压缩后能保持图像的特征不变，且在传递过程中可以抗干扰。

基于小波分析的图像压缩方法有很多，比较成功的有小波包最优基方法、小波域纹理模型方法、小波变换零树压缩、小波变换向量量化压缩等。

7.2.2 小波分析图像处理概述

1．图像的小波分解与重构

如图 7-4 所示是图像的小波分解示意图，L 表示低频，H 表示高频，下标 1、2 表示一级或二级分解。分解的数据传递示意图如图 7-5 所示（图中“↓2”表示下采样）。图 7-6 是重构的数据传递示意图（图中“↑2”表示上采样）。

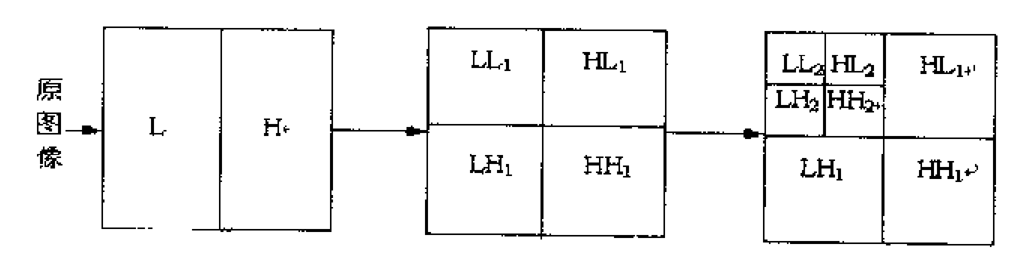


图 7-4 图像的小波分解示意图

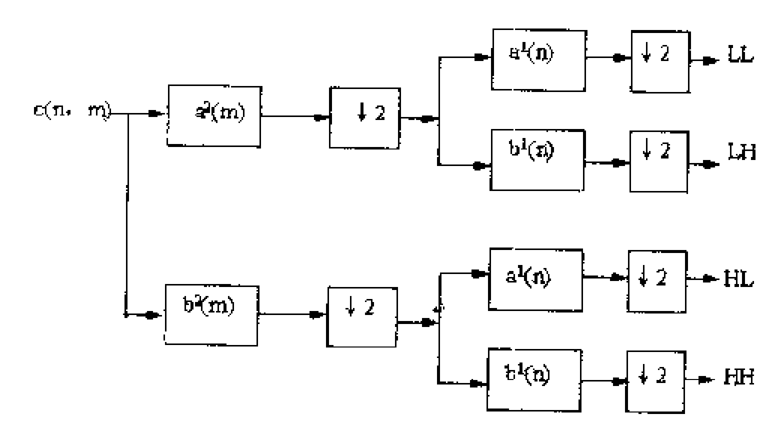


图 7-5 小波分解数据流示意图

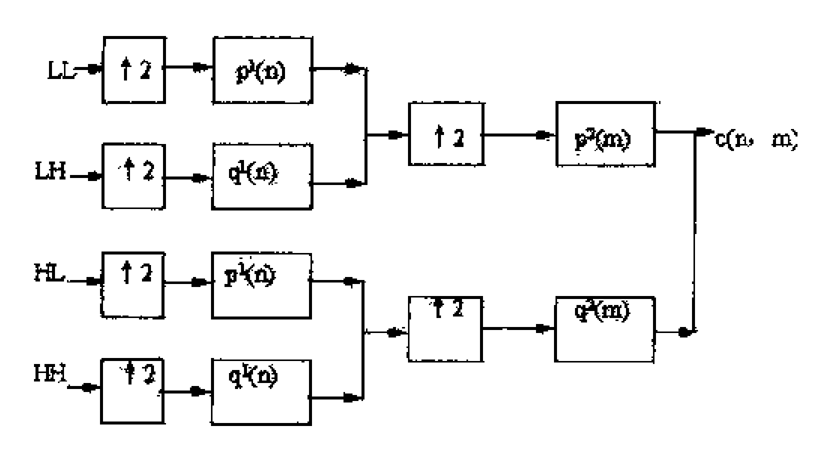


图 7-6 小波重构数据流示意图

这里我们简要地介绍一下上采样的概念。上采样是通过在样本之间插入零来增长信号分量的过程。如图 7-7 所示。

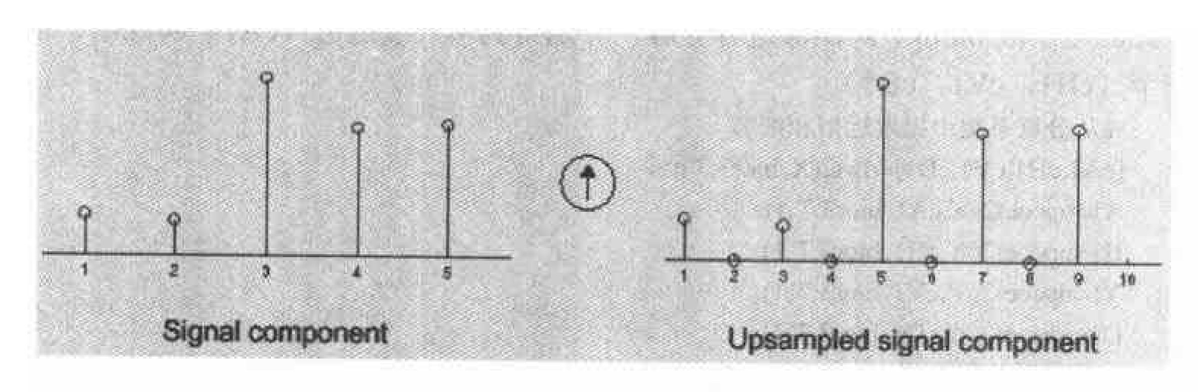


图 7-7 上采样示意图

【例 7-3】分析图像的小波分解和重构。

```
%从 MATLAB 命令行加载并显示图像
load wbarb;
whos
  Name        Size                 Bytes   Class

  X           256×256              double array
  map         192×3                4608    double array
image(X);
colormap(map);
colorbar;
```

结果如图 7-8 所示。

以下将其转换成为灰度级索引图像。

如果图像的色彩条是光滑的，那么小波变换就可以直接对其进行分解；否则需按前面所介绍的方法将图像转换成为灰度级索引图像方可进行分解。由图 7-8 可知，该索引图像的色彩条是光滑的，因此这里就可直接进行分解。

对图像进行单尺度分解：

```
[cA1,cH1,cV1,cD1]=dwt2(X,'bior3.7');
```

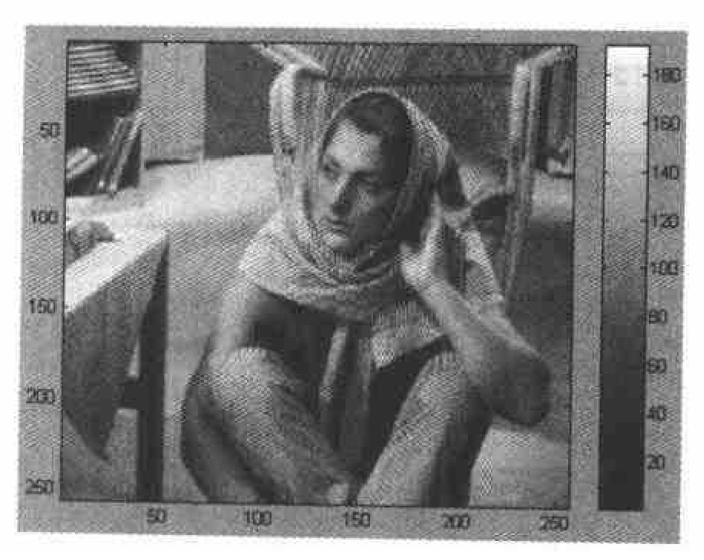


图 7-8　原始图像

在此应用小波 bior3.7 对图像进行分解，分别得到单尺度近似（cA1）和水平、垂直、对角细节（cH1，cV1，cD1）。

```
%从分解系数中提取近似和细节
[cA1,cH1,cV1,cD1]=dwt2(X,'bior3.7');
A1=upcoef2('a',cA1,'bior3.7',1);
H1=upcoef2('h',cH1,'bior3.7',1);
V1=upcoef2('v',cV1,'bior3.7',1);
D1=upcoef2('d',cD1,'bior3.7',1);
%显示近似和细节
colormap(map);
subplot(2,2,1);
image(wcodemat(A1,192));
title('近似 A1');
subplot(2,2,2);
image(wcodemat(H1,192));
title('水平细节 H1');
subplot(2,2,3);
image(wcodemat(V1,192));
title('垂直细节 V1');
subplot(2,2,4);
image(wcodemat(D1,192));
title('对角细节 D1');
```

结果如图 7-9 所示。

下面对图像进行多尺度分解。

同样地，对原始图像 X 利用小波 bior3.7 进行两层分解：

```
[C,S]=wavedec2(X,2,'bior3.7');
```

这里向量 C 中包含了两层分解后所有分量的系数，即第二层近似和前两层细节；向量 S 中保存了每一个分量的大小值。

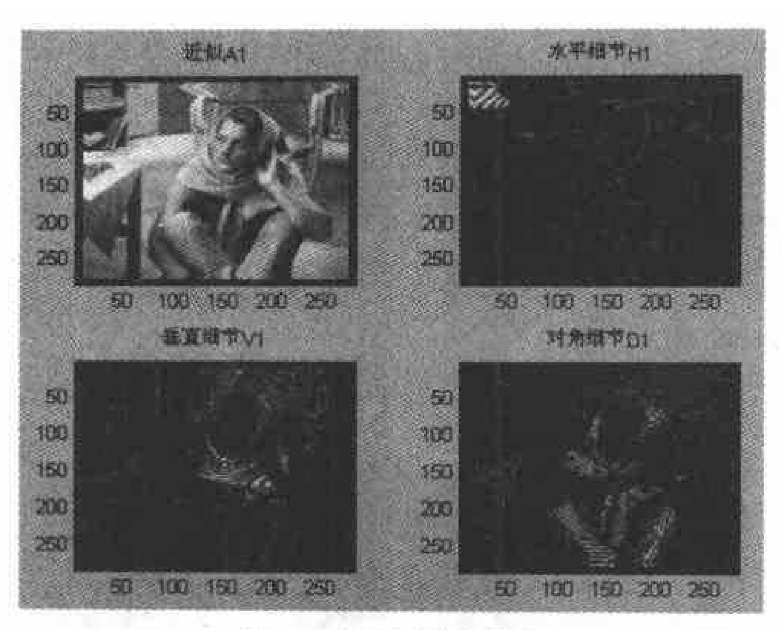


图 7-9　单尺度分解结果

```
%提取分解后的近似和细节系数
cA2=appcoef2(C,S,'bior3.7',2);
[cH2,cV2,cD2]=detcoef2('all',C,S,2);
[cH1,cV1,cD1]=detcoef2('all',C,S,2);
%从系数 C 重构第二层近似
A2=wrcoef2('a',C,S,'bior3.7',2);
%从系数 C 重构第一、二层细节
H1=wrcoef2('h',C,S,'bior3.7',1);
V1=wrcoef2('v',C,S,'bior3.7',1);
D1=wrcoef2('d',C,S,'bior3.7',1);
H2=wrcoef2('h',C,S,'bior3.7',2);
V2=wrcoef2('v',C,S,'bior3.7',2);
D2=wrcoef2('d',C,S,'bior3.7',2);
%显示多尺度分解的结果
colormap(map);
subplot(2,4,1);
image(wcodemat(A1,192));
title('近似 A1');
subplot(2,4,2);
image(wcodemat(H1,192));
title('水平细节 H1');
subplot(2,4,3);
image(wcodemat(V1,192));
title('垂直细节 V1');
subplot(2,4,4);
image(wcodemat(D1,192));
title('对角细节 D1');
subplot(2,4,5);
image(wcodemat(A2,192));
title('近似 A2');
```

```
subplot(2,4,6);
image(wcodemat(H2,192));
title('水平细节 H2');
subplot(2,4,7);
image(wcodemat(V2,192));
title('垂直细节 V2');
subplot(2,4,8);
image(wcodemat(D2,192));
title('对角细节 D2');
```

结果如图 7-10 所示。

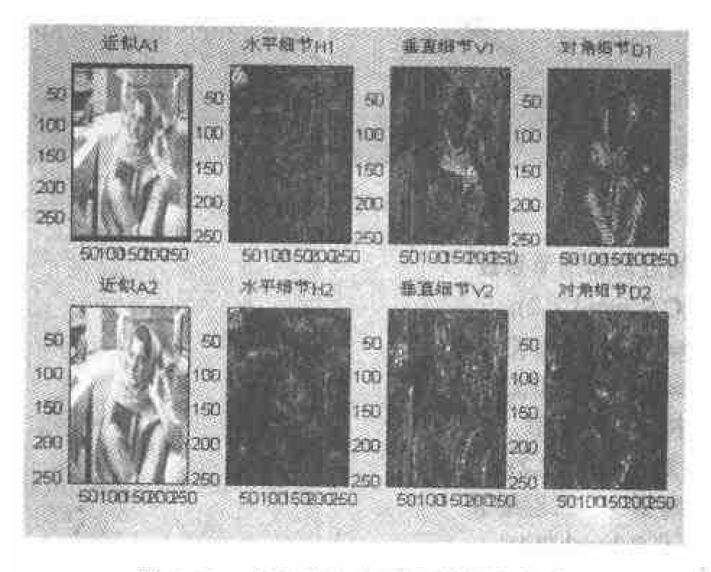


图 7-10　多尺度分解后的近似和细节

```
%从多尺度分解后的系数重构原始图像并显示结果
X0=waverec2(C,S,'bior3.7');
image(X0);
colormap(map);
colorbar;
```

结果如图 7-11 所示。

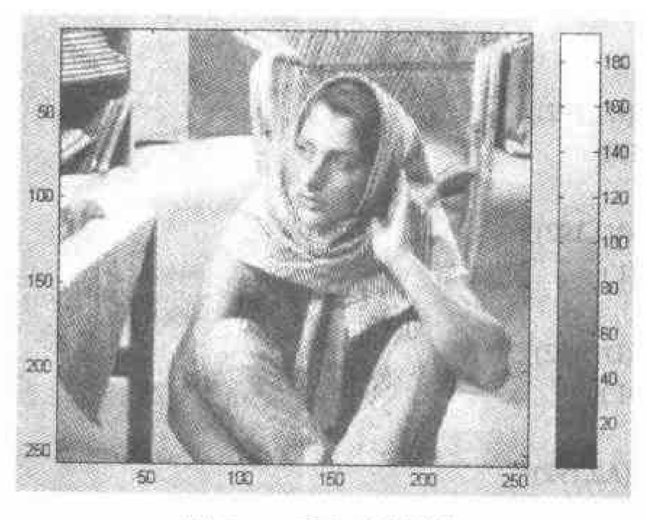


图 7-11　重构后的图像

本例程序清单如例程 7-3。

例程 7-3

```
%从 MATLAB 命令行加载并显示图像
load wbarb;
whos
  Name        Size            Bytes    Class

  X         256×256              double array
  map       192×3               4608    double array
image(X);
colormap(map);
colorbar;
==============================
%转换成为灰度级索引图像
%对图像进行单尺度分解
[cA1,cH1,cV1,cD1]=dwt2(X,'bior3.7');
==============================
%从分解系数中提取近似和细节
[cA1,cH1,cV1,cD1]=dwt2(X,'bior3.7');
A1=upcoef2('a',cA1,'bior3.7',1);
H1=upcoef2('h',cH1,'bior3.7',1);
V1=upcoef2('v',cV1,'bior3.7',1);
D1=upcoef2('d',cD1,'bior3.7',1);
==============================
%显示近似和细节
colormap(map);
subplot(2,2,1);
image(wcodemat(A1,192));
title('近似 A1');
subplot(2,2,2);
image(wcodemat(H1,192));
title('水平细节 H1');
subplot(2,2,3);
image(wcodemat(V1,192));
title('垂直细节 V1');
subplot(2,2,4);
image(wcodemat(D1,192));
title('对角细节 D1');
==============================
%对图像进行多尺度分解
[C,S]=wavedec2(X,2,'bior3.7');
==============================
%提取分解后的近似和细节系数
cA2=appcoef2(C,S,'bior3.7',2);
[cH2,cV2,cD2]=detcoef2('all',C,S,2);
```

```
[cH1,cV1,cD1]=detcoef2('all',C,S,2);
==================================
%从系数 C 重构第二层近似
A2=wrcoef2('a',C,S,'bior3.7',2);
==================================
%从系数 C 重构第一、二层细节
H1=wrcoef2('h',C,S,'bior3.7',1);
V1=wrcoef2('v',C,S,'bior3.7',1);
D1=wrcoef2('d',C,S,'bior3.7',1);
H2=wrcoef2('h',C,S,'bior3.7',2);
V2=wrcoef2('v',C,S,'bior3.7',2);
D2=wrcoef2('d',C,S,'bior3.7',2);
==================================
%显示多尺度分解的结果
colormap(map);
subplot(2,4,1);
image(wcodemat(A1,192));
title('近似 A1');
subplot(2,4,2);
image(wcodemat(H1,192));
title('水平细节 H1');
subplot(2,4,3);
image(wcodemat(V1,192));
title('垂直细节 V1');
subplot(2,4,4);
image(wcodemat(D1,192));
title('对角细节 D1');
subplot(2,4,5);
image(wcodemat(A2,192));
title('近似 A2');
subplot(2,4,6);
image(wcodemat(H2,192));
title('水平细节 H2');
subplot(2,4,7);
image(wcodemat(V2,192));
title('垂直细节 V2');
subplot(2,4,8);
image(wcodemat(D2,192));
title('对角细节 D2');
==================================
%从多尺度分解后的系数重构原始图像并显示结果
X0=waverec2(C,S,'bior3.7');
image(X0);
colormap(map);
colorbar;
```



2. 图像边缘失真的处理

为了处理信号的边界失真问题，应当有区别对待信号的边界和信号的其他部分。下面介绍几种延拓信号的方法。这些方法有：补零（Zero Padding）、平滑填补（Smooth Padding）、周期延拓（Periodic Extension）、边界值复制或边界对称化（Boundary Value Replication 或 Symmetrization）等。

通常，在边界上使用基于信号延拓的简单方法是行之有效的。这里包括在分解过程中的每一级都需计算一些额外的系数，因此应当注意在分解过程中的每一级延拓是必须的。

（1）补零（Zero.Padding）

这种方法假设在原始支撑以外的信号以零补足。其缺点是人为地在边界处制造了不连续性。

（2）边界对称化（Symmetrization）

这种方法假设通过对称的边界值复制可以恢复信号和图像的原始支撑以外的信号和图像。这种方法也是小波工具箱中小波变换的默认模式。其缺点是在边界处，人为地制造了一阶导数的不连续性，但是这种方法通常对图像处理非常有效。

（3）一阶平滑填补（Smooth Padding of Order 1）

在这个方法中，假设通过简单的一阶导数外插（填补时，对前两个值和后两个值使用线性拟合）能够从信号或图像的原始支撑之外恢复信号或图像。该方法通常对光滑信号较为有效。

（4）零阶平滑填补（Smooth Padding of Order 0）

在这个方法中，假设通过简单的常数外插能够从信号或图像的原始支撑之外恢复信号或图像。对于信号延拓来说，该方法是位于左边的第一个值和右边的最后一个值的重复。

（5）周期性填补（1）（Periodic Padding (1)）

这里假设通过周期延拓恢复信号或图像原始支撑以外的信号或图像。其缺点是在边界处人为地制造了不连续性。

在上述5种延拓模式中，存在一定的冗余，因此，对任意一种延拓模式，在逆变换中，都能确保对信号和图像的完全重构。

（6）周期性填补（2）（Periodic Padding(2)）

如果信号的长度是奇数，首先给信号加一个采样点，其值等于最后一个值，接下来，对信号进行周期性填补（1），即在两端对信号进行最小周期延拓。对于图像，采取同样的方法。这种模式可生成最小长度的小波分解，但是为了确保完全重构，在逆变换中也应采用同样的延拓模式。

让我们比较一下理论离散小波变换（Discrete Wavelet Transform，DWT）和实际离散小波变换的某些性质。

理论DWT应用于信号分析时，信号是在无限长度的时间区间上。对于正交小波，DWT具有如下特性：

- 保范性

令 cA 和 cD 分别是无限长信号 X 的DWT系数的近似和细节，那么它们之间有如下关系：

$$X^2 = cA^2 + cD^2$$

- 正交性

令 A 和 D 分别是重构近似和细节，那么，A 和 D 是正交的，即：

$$X^2 = A^2 + D^2$$

- 完全重构

$$X = A + D$$

由于 DWT 是用于无限长时间区间的信号，而分解需要对信号进行延拓，因此，在重构时需要进行截断处理。对任意选择的信号长度、小波和延拓模式，为了保证关键的性质完全重构，从而使得保范性和正交性不能满足。这些特性对于那些比原始信号长度大的延拓信号是可以保证的，所以仅有完全重构性对任意信号、小波和延拓模式总是可以保证的。如果使用周期延拓模式进行离散小波变换且信号长度可被 2^J（J 是最大的小波分解层数）除尽，那么上述三条性质均可得到保证。

【例 7-4】图像边缘失真处理。

```
%给定一个原始图像
load geometry;
nbcol=size(map,1);
colormap(pink(nbcol));
image(wcodemat(X,nbcol));
title('原始图像');
```

结果如图 7-12 所示。

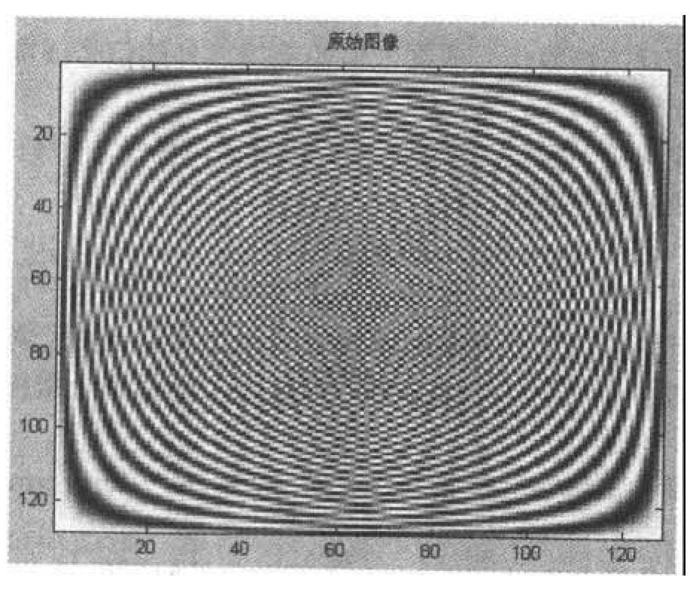


图 7-12 原始图像

现在设置延拓模式为补零方式，对图像应用小波 sym4 进行 3 层分解，然后重构第 3 层近似。

```
%补零延拓
```

```
lev=3;
wname='sym4';
dwtmode('zpd');
[c,s]=wavedec2(X,lev,wname);
a=wrcoef2('a',c,s,wname,lev);
image(wcodemat(a,nbcol));
```

结果如图 7-13 所示。

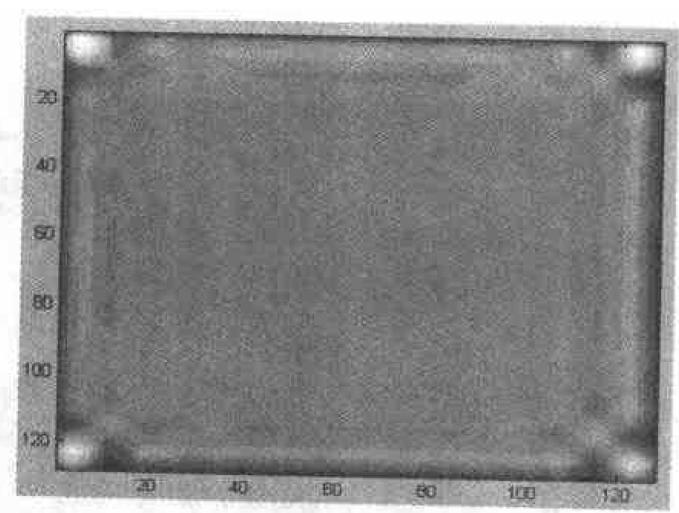


图 7-13　补零延拓模式的第 3 层重构近似

在此应注意，MATLAB 6.5 提示了默认延拓模式被改变，并给出了所用延拓模式。

下面我们改变延拓模式为对称延拓且同样应用小波 sym4 将原始图像分解到第 3 层，之后重构第 3 层近似。

```
dwtmode('sym');
[c,s]=wavedec2(X,lev,wname);
aa=wrcoef2('a',c,s,wname,lev);
image(wcodemat(aa,nbcol));
```

结果如图 7-14 所示。

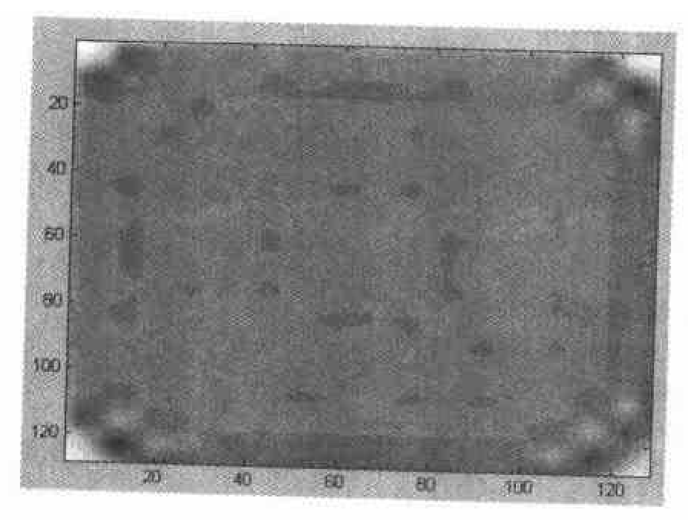


图 7-14　对称延拓模式的第 3 层重构近似

最后，我们设置平滑填补的延拓模式，应用小波 sym4 对原始图像进行 3 层分解，然后重构第 3 层近似。

```
dwtmode('spd');
[c,s]=wavedec2(X,lev,wname);
aaa=wrcoef2('a',c,s,wname,lev);
image(wcodemat(aaa,nbcol));
```

结果如图 7-15 所示。

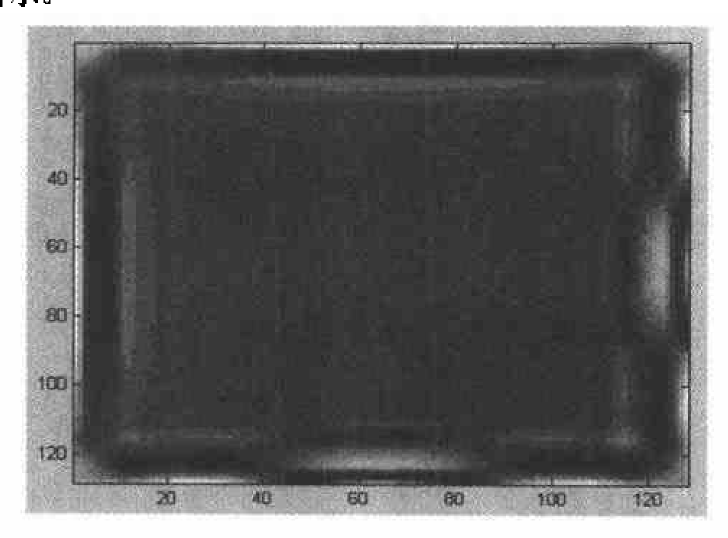


图 7-15　平滑延拓模式的第 3 层重构近似

本例程清单如例程 7-4。

例程 7-4

```
%给定一个原始图像
load geometry;
nbcol=size(map,1);
colormap(pink(nbcol));
image(wcodemat(X,nbcol));
title('原始图像');
```

```
%设置延拓模式为补零方式，对图像应用小波 sym4 进行 3 层分解，然后重构第 3 层近似
lev=3;
wname='sym4';
dwtmode('zpd');
[c,s]=wavedec2(X,lev,wname);
a=wrcoef2('a',c,s,wname,lev);
image(wcodemat(a,nbcol));
```

```
%改变延拓模式为对称延拓且同样应用小波 sym4 将原始图像分解到第 3 层，之后重构第 3 层近似
dwtmode('sym');
[c,s]=wavedec2(X,lev,wname);
```

```
aa=wrcoef2('a',c,s,wname,lev);
image(wcodemat(aa,nbcol));
=============================
%设置平滑填补的延拓模式，应用小波 sym4 对原始图像进行 3 层分解，然后重构第 3 层近似
dwtmode('spd');
[c,s]=wavedec2(X,lev,wname);
aaa=wrcoef2('a',c,s,wname,lev);
image(wcodemat(aaa,nbcol));
```

7.3　二维小波分析用于图像处理

7.3.1　图像消噪

1．小波分析进行图像消噪

二维图像信号的消噪步骤有三步，和一维信号的消噪步骤基本相同，所不同的只是处理时用二维小波分析工具代替了一维小波分析工具。因此，对二维图像信号的消噪方法也同样适合于一维信号，尤其对几何图形更为适合。二维小波分析用于图像消噪的步骤为：

（1）二维图像信号的小波分解

在这一步，应当选择合适的小波和恰当的分解层次（记为 N），然后对待分析的二维图像信号 X 进行 N 层分解计算。

（2）对分解后的高频系数进行阈值量化

对于分解的每一层，选择一个恰当的阈值，并对该层高频系数进行软阈值量化处理。在此，阈值选取规则同前面的信号处理部分。

（3）二维小波的重构图像信号

同样地，根据小波分解后的第 N 层近似（低频系数）和经过阈值量化处理后的各层细节（高频系数），来计算二维信号的小波重构。

在这三步中，重点内容仍然是如何选取阈值和如何进行阈值量化这个步骤。下面让我们用具体的例子来说明利用小波分析进行图像消噪这个问题。

【例 7-5】给定一个二维含噪图像，请利用小波分析对其进行消噪处理。

例程 7-5

```
%装载原始图像信号并图示
load wgatlin;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;
=============================
%生成含噪图像并图示
```

```
init=2055615866;
rand('seed',init);
XX=X+12*randn(size(X));
subplot(2,2,2);
image(XX);
colormap(map);
title('含噪图像');
axis square;
==================================
%下面对图像消噪
%首先用 sym4 小波函数对 XX 进行 2 层分解
[c,l]=wavedec2(XX,2,'sym4');
%实现低通滤波消噪
a1=wrcoef2('a',c,l,'sym4',1);
%再次实现低通滤波消噪
a2=wrcoef2('a',c,l,'sym4',2);
==================================
%图示消噪处理后的结果
subplot(2,2,3);
image(a1);
colormap(map);
title('第一次消噪图像');
axis square;
subplot(2,2,4);
image(a2);
colormap(map);
title('第二次消噪图像');
axis square;
```

结果如图 7-16 所示。

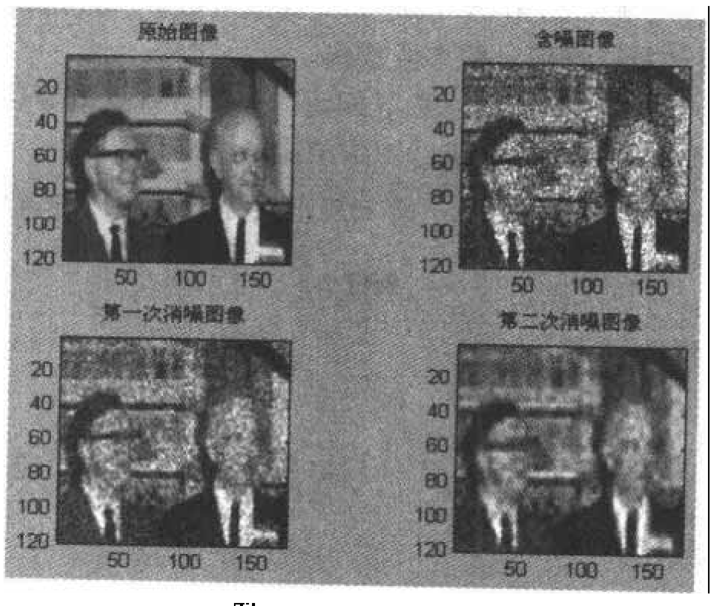


图 7-16 运行结果

【例 7-6】利用二维小波变换对给定图像 gatlin2.mat 进行小波消噪处理。

例程 7-6

```
%装载并图示原始图像
load gatlin2;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;
==============================
%生成含噪图像并图示
init=2055615866;
randn('seed',init);
XX=X+8*randn(size(X));
subplot(2,2,2);
image(XX);
colormap(map);
title('含噪图像');
axis square;
==============================
%对图像进行消噪处理
%用小波函数 coif2 对图像 XX 进行 2 层分解
[c,l]=wavedec2(XX,2,'coif2');
n=[1,2];%设置尺度向量
p=[10.28,24.08];%设置阈值向量
==============================
%对三个高频系数进行阈值处理
nc=wthcoef2('h',c,l,n,p,'s');
nc=wthcoef2('v',c,l,n,p,'s');
nc=wthcoef2('d',c,l,n,p,'s');
X1=waverec2(nc,l,'coif2');
subplot(2,2,3);
image(X1);
colormap(map);
title('第一次消噪后的图像');
axis square;
    ===========================
%再次对三个高频系数进行阈值处理
mc=wthcoef2('h',nc,l,n,p,'s');
mc=wthcoef2('v',nc,l,n,p,'s');
mc=wthcoef2('d',nc,l,n,p,'s');
==========================================
%对更新后的小波分解结构进行重构并图示结果
X2=waverec2(mc,l,'coif2');
subplot(2,2,4);
```

```
image(X2);
colormap(map);
title('第二次消噪后的图像');
axis square;
```

结果如图 7-17 所示。

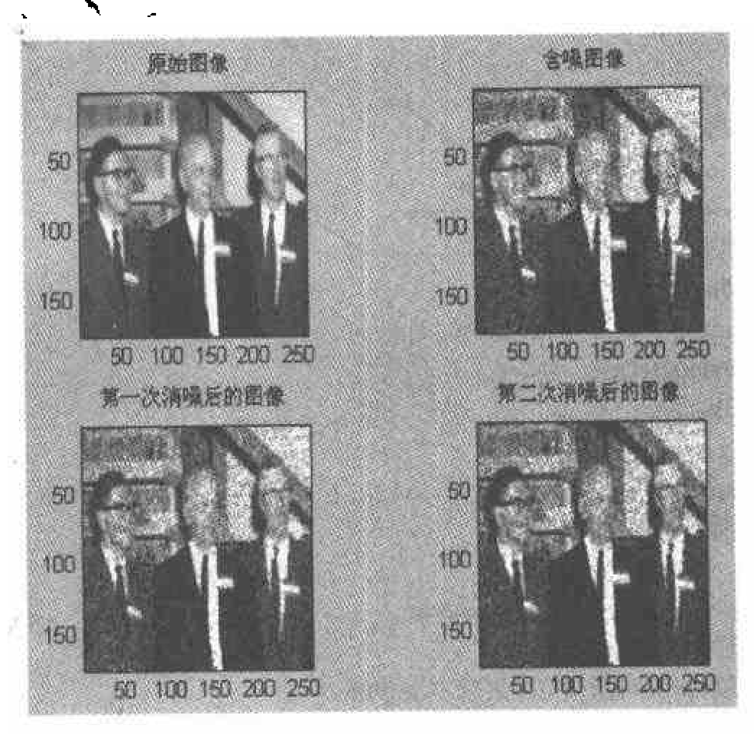


图 7-17 运行结果

2. 小波包分析进行图像消噪

在本节中，小波包分析进行图像消噪与压缩处理的基本原理和方法与前一章中所介绍的对信号消噪与压缩的相同，请读者朋友参看第 6 章的相关内容。本节仅以具体的实例来说明小波包分析在图像消噪处理中的应用。

【例 7-7】给定一个二维含噪图像，请利用小波包分解对其进行消噪处理。

例程 7-7

```
%装载并图示原始图像
load gatlin2;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;
==============================
%生成含噪图像并图示
init=2055615866;
randn('seed',init);
```

```
X1=X+10*randn(size(X));
subplot(2,2,2);
image(X1);
colormap(map);
title('含噪图像');
axis square;
========================================
%消噪处理：设置函数 wpdencmp 的消噪参数
thr=10;sorh='s';
crit='shannon';
keepapp=0;
X2=wpdencmp(X1,sorh,3,'sym4',crit,thr,keepapp);
==============================
%画出消噪后的图像
subplot(2,2,3);
image(X2);
colormap(map);
title('全局阈值消噪图像');
axis square;
=======================================================
%对图像进行平滑处理以增强消噪效果，这里采用中值滤波的方法
for i=2:175;
    for j=2:259
        Xtemp=0;
        for m=1:3
            for n=1:3
                Xtemp=Xtemp+X2(i+m.2,j+n.2);
            end
        end
        Xtemp=Xtemp/9;
        X3(i,j)=Xtemp;
    end
end
==============================
%图示平滑结果
subplot(2,2,4);
image(X3);
colormap(map);
title('平滑后的图像');
axis square;
```

结果如图 7-18 所示。

【例 7-8】利用小波包分解，设置阈值，对给定含噪图像进行消噪处理。

解决思想是首先生成题设条件的含噪图像，然后对其进行相应的小波包分解，最后对它的低频部分进行消噪处理后重构图像。

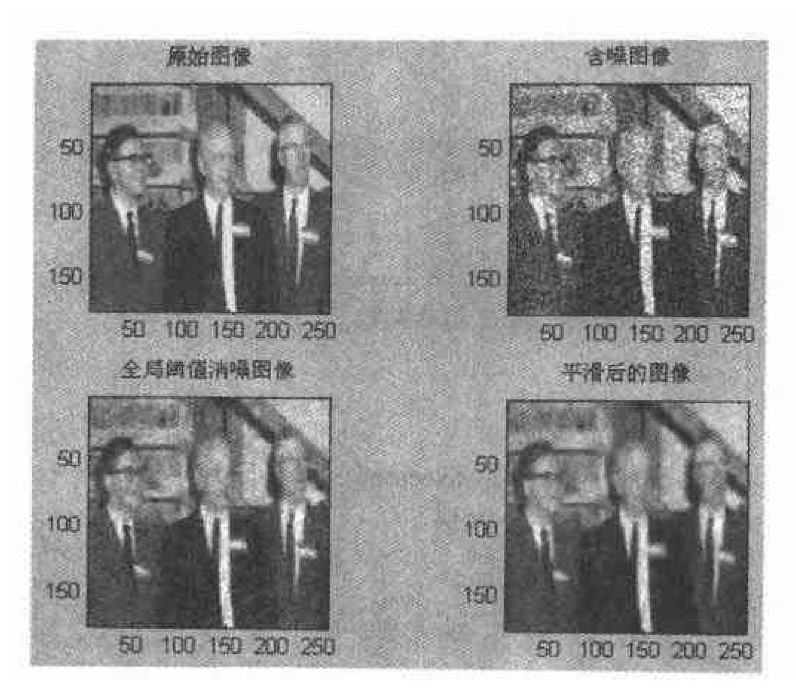


图 7-18 运行结果

例程 7-8

```
%装载并图示原始图像
load flujet;
subplot(2,2,1);
image(X);
colormap(map);
title('原始图像');
axis square;
===================================
%生成含噪图像并图示
init=2055615866;
randn('seed',init);
X1=X+20*randn(size(X));
subplot(2,2,2);
image(X1);
colormap(map);
title('含噪图像');
axis square;
===================================
%用小波 sym2 对图像 X1 进行一层小波包分解
T=wpdec2(X1,1,'sym2');
===================================
%设置全局阈值
thr=8.342;
===================================
%对图像的小波包分解系数进行软阈值量化以实现图像消噪
```

```
NT=wpthcoef(T,0,'s',thr);
%由于所含的是高频白噪声，因此仅对低频系数进行重构
X2=wprcoef(NT,1);
%画出结果图像
subplot(2,2,3);
image(X2);
colormap(map);
title('消噪后的图像');
axis square;
```

结果如图 7-19 所示。

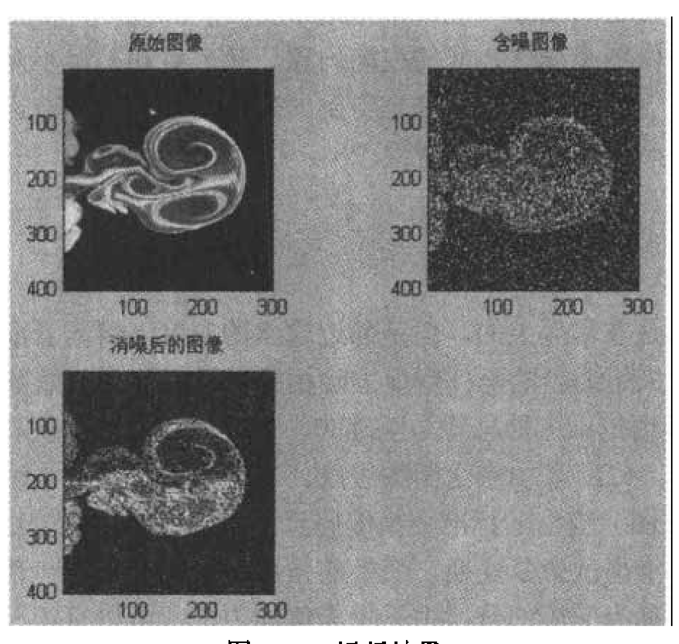


图 7-19　运行结果

至此，我们已经向读者介绍了小波及小波包分析用于一维信号（第 6 章内容）和二维信号（图像）的处理方面内容。在以上的介绍中，我们可以发现，小波变换在非线性处理领域得到了广泛应用。在传统的基于傅里叶变换的信号处理方法中，要使信号和噪声的频带重叠部分尽可能地小，这样，在频域才可通过不变滤波的方法将信号和噪声区分开；而当它们的频域重叠时，这种方法就无能为力了。但是如果采用线性小波变换和其他时间频率（或时间–尺度）法，通过选择不同基的方法使在相应坐标系内的信号和噪声的重叠尽可能地小，从而可进行信噪分离。但对大多数信号来说，合适的基的选择就是一个难题，因此这种方法的应用受到了限制。

而我们常用的是基于离散小波变换的非线性滤波方法，在这种方法中，尽管谱可以任意重叠，但是谱的幅度是尽可能不同的。因此这种方法允许对变换系数进行切削、阈值处理、缩小幅度范围等来分离信号和消噪。这种非线性滤波方法之所以特别有效是由于小波变换具有一种“集中”的能力，因此可以使一个信号的能量在小波变换域集中于少数系数

上，那么相对来说，这些系数的取值必然大于在小波变换域内能量分散于大量小波系数上的信号或噪声的小波系数值。这就意味着对小波系数进行阈值处理可以在小波变换域中除去低幅度的噪声和我们所不期望的信号，然后再进行逆离散小波变换即可。尽管所恢复的信号丢失了一点细节，但是仍能恢复我们所期望的信号。这种方法不仅可以用于信号消噪，在数据压缩和分离信号方面也极其有用。

另外，我们对信号消噪和压缩处理中所用到的软阈值和硬阈值做一简单讨论。在消噪和压缩处理中，如果采用软阈值处理方法，则估计信号和原始信号具有同样的光滑性，这是因为小波是一类光滑函数的无条件基，并且软阈值处理保证满足缩小条件，正是缩小条件保证了估计函数和原始函数具有相同的光滑性，此外，软阈值处理方法是满足缩小条件的最优估计。这种光滑性保证了软阈值处理方法不同于硬阈值或其他常用的频域处理方法，它所得到的估计信号同原始信号相比不会产生附加振荡。和传统的方法相似，阈值处理方法也会造成小程度上的图像（或数据）细节丢失。因此，在使用该方法时，还要考虑抑制噪声与所保留图像细小变化之间的权衡问题。从范数误差最小观点来说，硬阈值处理方法优于软阈值方法。但是由硬阈值处理得到的估计信号通常会有许多不期望的振荡，并且不具有期望的同原始信号相同的光滑性。

7.3.2 图像压缩

虽然图像的数据是非常巨大的，但是邻近像素的灰度（将其看成随机变量）往往是高度相关的。因而可以采用适当的坐标变换去除相关，从而达到压缩数据的目的。

传统的 K.L 变换也是以这种思想为基础的，它把信号的一小块看成是一个独立的随机向量，在这种模型下，K.L 基由余弦函数组成。由联合图像专家小组（JPEG，图 7-20 给出了 JPEG 的压缩框图）提出的标准变换编码算法中包含将图像分成 8×8 像素的方块，然后在方块内进行二维离散余弦变换。变换的系数分成两部分，一部分是零频率，称为直流部分，另一部分则称为交流部分。由于像素值均为正，所以直流部分是较大的正值，且在块与块间缓慢地变化，可以高精度地量化。因此直流部分可以有效地代表缓变部分，对交流部分根据视觉特性赋予一定的权值，这种视觉特性根据人的主观感觉通过试验决定，量化时还需考虑到一定的压缩率。

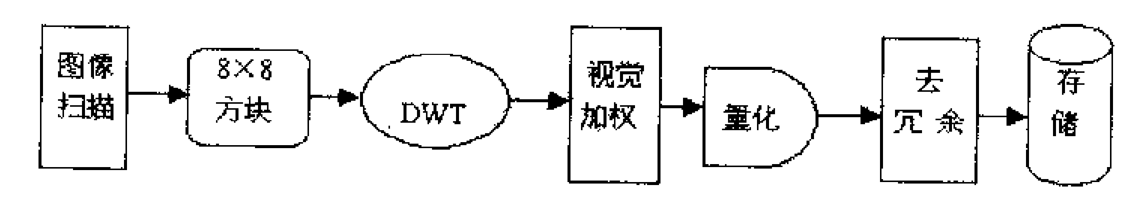


图 7-20　JPEG 的压缩框图

JPEG 及其变种是按空间进行分段的。我们也可以按尺度分段，或按尺度和频率分段后选择一种最优的表示，这可通过小波变换实现。

小波或多分辨分析方法将一幅图像分成近似和细节两部分，细节对应的是小尺度的瞬变，但在本尺度内看起来很稳定。因此将细节存储起来，对近似部分在下一个尺度上进行分解，重复该过程即可。图 7-21 给出了本算法的框图。

图 7-21　小波变换图像压缩

近似与细节在正交镜像滤波器算法中分别对应于高通和低通滤波，这种变换通过尺度去掉相关性，在视频压缩中证明是有效的。这种方法有一个优点，即图像分成按细节增多的层，因此可以先给出一幅较为粗糙的图像，然后可根据需要提供更好的细节。

小波包包含很多基，并且能在一给定的尺度下，提供一种更好的分析频率成分瞬变的方法。这一点可通过对近似与细节都进行更进一步的滤波，结果先保存下来，然后按某种代价函数寻找最优基来实现。下面给出利用小波包基及小波包最优基进行图像压缩的算法框图，如图 7-22 和图 7-23 所示。

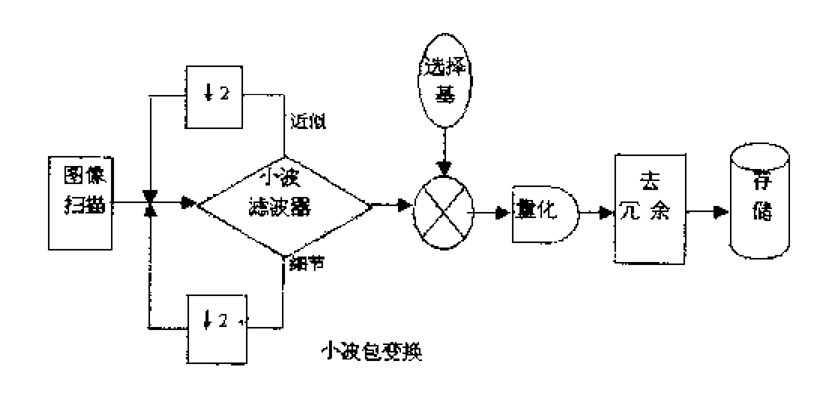


图 7-22　利用小波包基进行图像压缩

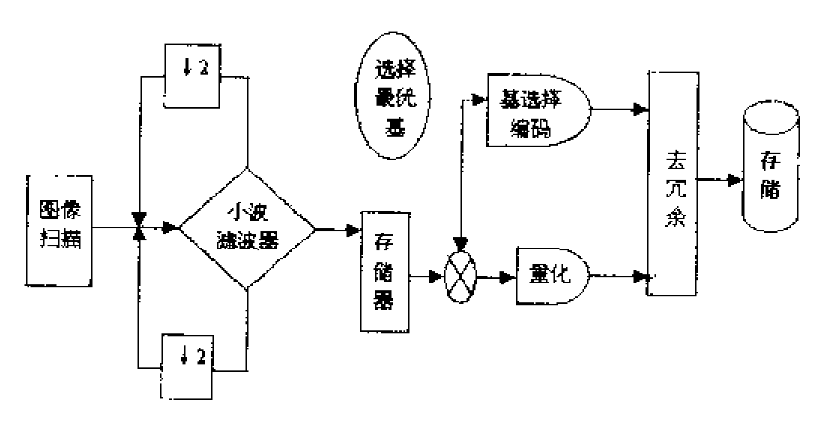


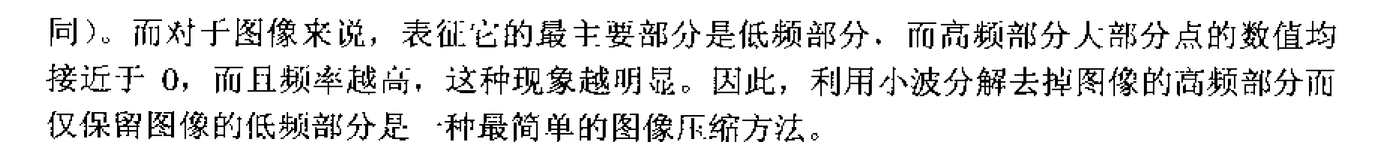
图 7-23　利用小波包最优基进行图像压缩

下面举几个图像压缩的实例。

1．小波分析进行图像压缩

【例 7-9】利用二维小波分析对给定图像进行压缩。

对图像做小波分解后，可得到一系列不同分辨率的子图像（它们所对应的频率不相



同）。而对于图像来说，表征它的最主要部分是低频部分，而高频部分大部分点的数值均接近于 0，而且频率越高，这种现象越明显。因此，利用小波分解去掉图像的高频部分而仅保留图像的低频部分是一种最简单的图像压缩方法。

例程 7-9

```
%装载并显示原始图像
load woman;
subplot(2,2,1);
image(X);colormap(map);
title('原始图像 ');
axis square;
disp('压缩前图像的大小：');
whos('X')
=============================
%分解图像，提取分解结构中的第一层系数
[c,l]=wavedec2(X,2,'bior3.7');
cA1=appcoef2(c,l,'bior3.7',1);
cH1=detcoef2('h',c,l,1);
cD1=detcoef2('d',c,l,1);
cV1=detcoef2('v',c,l,1);
=============================
%重构第一层系数
A1=wrcoef2('a',c,l,'bior3.7',1);
H1=wrcoef2('h',c,l,'bior3.7',1);
D1=wrcoef2('d',c,l,'bior3.7',1);
V1=wrcoef2('v',c,l,'bior3.7',1);
c1=[A1 H1;V1 D1];
=============================
%图示第一层各频率信息
subplot(2,2,2);
image(c1);
title('分解后的低频和高频信息');
axis square;
=============================================
%对图像进行压缩：保留第一层低频信息并对其进行量化编码
ca1=wcodemat(cA1,440,'mat',0);
ca1=0.5*ca1;
subplot(2,2,3);
image(ca1);
colormap(map);
title('第一次压缩图像');
axis square;
disp('第一次压缩图像的大小：');
whos('ca1')
=========================================
%压缩图像：保留第二层低频信息并对其进行量化编码
```

```
cA2=appcoef2(c,l,'bior3.7',2);
ca2=wcodemat(cA2,440,'mat',0);
ca2=0.5*ca2;
subplot(2,2,4);
image(ca2);
colormap(map);
title('第二次压缩图像');
axis square;
disp('第二次压缩图像大小: ');
whos('ca2')
```

首先给出压缩后图像的大小，从所给结果中可以看出，随着分解层数的增加，压缩比是递减的。压缩后的图像如图 7-24 所示。

压缩前图像的大小：

```
Name        Size          Bytes    Class
X         256×256        524288   double array
```

第一次压缩图像的大小：

```
Name        Size          Bytes    Class
ca1       135×135        145800   double array
```

第二次压缩图像大小

```
Name        Size          Bytes    Class
ca2        75×75          45000   double array
```

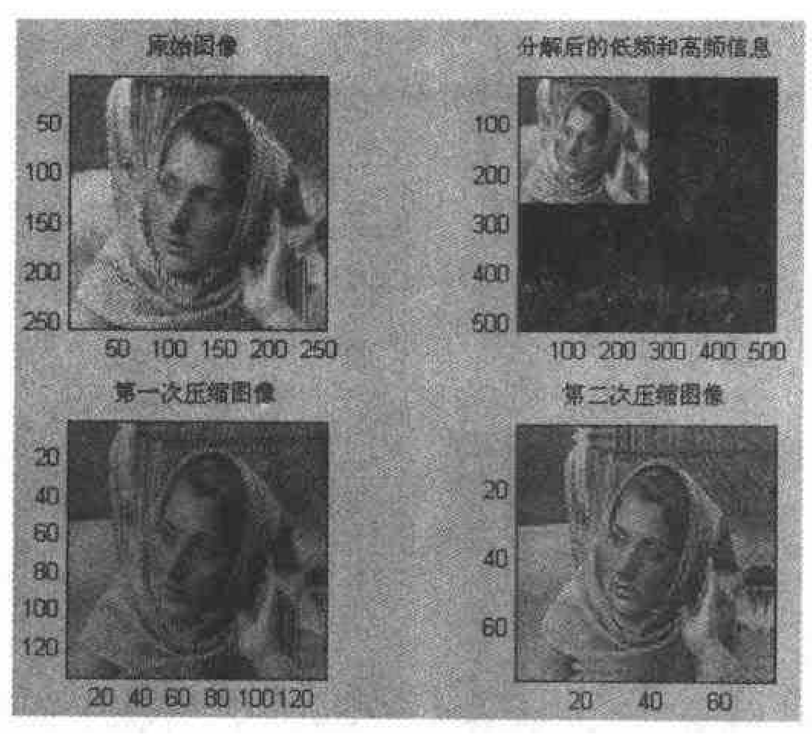


图 7-24 运行结果

在 MATLAB 的小波工具箱中，有专门用于压缩处理的函数 wdencmp。下面我们就通过一个具体的例子来说明该函数的应用及其特点。

【例 7-10】利用函数 wdencmp 对给定图像进行压缩处理。

例程 7-10

```
%装载并显示原始图像
load flujet;
subplot(1,2,1);
image(X);colormap(map);
title('原始图像');
axis square;
==============================
%首先利用小波 db3 对图像 X 进行 2 层分解
[c,l]=wavedec2(X,2,'db3');
==============================
%全局阈值
[thr,sorh,keepapp]=ddencmp('cmp','wv',X);
==============================
%进行压缩处理：对所有高频系数进行同样的阈值量化处理
[Xcmp,cxc,lxc,perf0,perfl2]=wdencmp('gbl',c,l,'db3',2,thr,sorh,keepapp);
==============================
%图示压缩结果
subplot(1,2,2);
image(Xcmp);colormap(map);
title('压缩后的图像');
axis square;
disp('小波分解系数中置 0 的系数个数百分比：');
perf0
disp('压缩后保留能量百分比：');
perfl2
```

首先给出应用函数 wdencmp 进行压缩的比率，即分解系数中置 0 的系数个数百分比和保留能量百分比。压缩图像如图 7-25 所示。

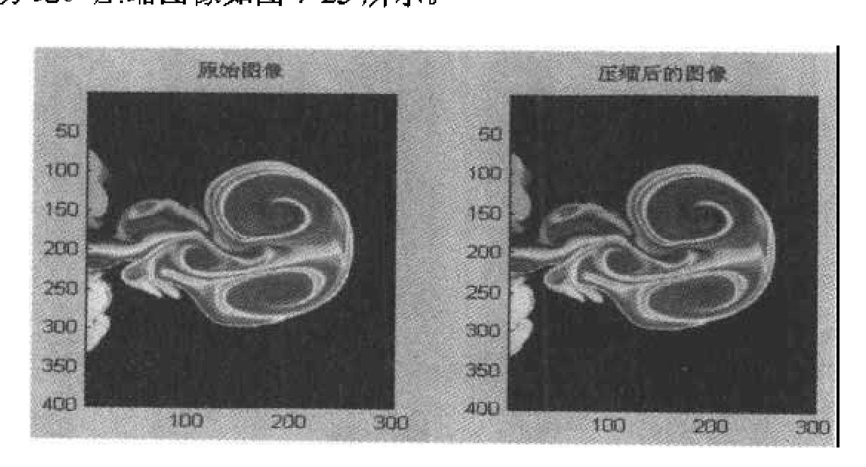


图 7-25　运行结果

小波分解系数中置 0 的系数个数百分比：

```
perf0 =
    89.5829
```

压缩后保留能量百分比：

```
perfl2 =
99.9831
```

2．小波包分析进行图像压缩

在 MATLAB 的小波工具箱中，提供了一个函数 wpdencmp，这个函数是专门用小波包进行信号的消噪和压缩处理的。这里我们通过具体的例子来看一下它的压缩功能。

【例 7-11】利用小波包分析对给定图像进行压缩处理。

例程 7-11

```
%装载并显示原始图像
load wbarb;
subplot(1,2,1);
image(X);colormap(map);
title('原始图像');
================================
%采用默认的全局阈值对图像进行压缩
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',X);
Xc=wpdencmp(X,sorh,3,'bior3.1',crit,thr,keepapp);
subplot(1,2,2);
image(Xc);colormap(map);
title('全局阈值压缩图像');
```

结果如图 7-26 所示。

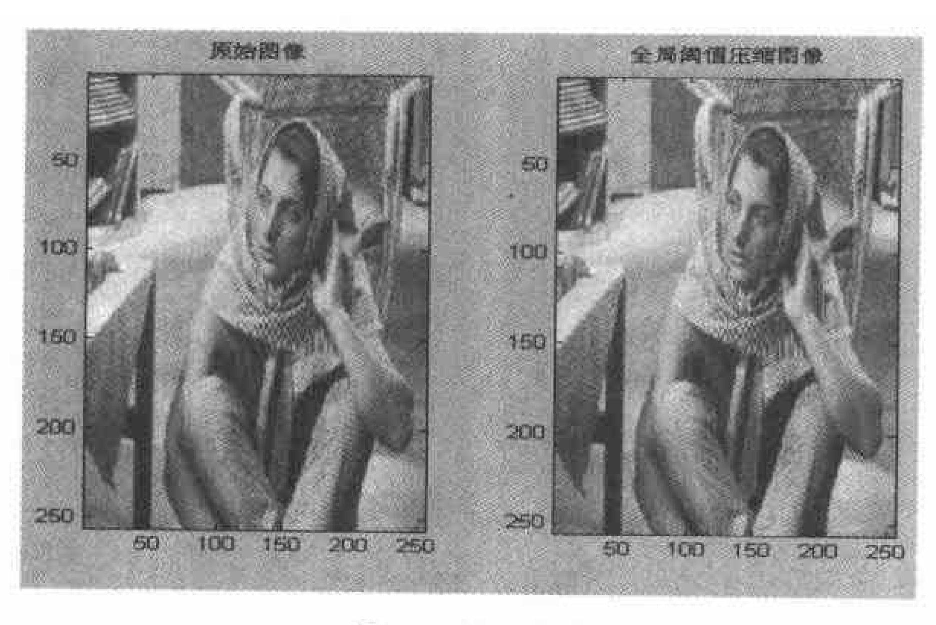


图 7-26　运行结果

【例 7-12】利用小波包进行图像压缩处理，并观察其压缩效率。

例程 7-12

```
%装载并显示原始图像
load detfingr;
subplot(1,2,1);
image(X);colormap(map);
title('原始图像');
================================
%采用默认的全局阈值进行图像压缩处理
[thr,sorh,keepapp,crit]=ddencmp('cmp','wp',X);
[Xc,treed,perf0,perfl2]=wpdencmp(X,sorh,3,'bior3.1',crit,thr,keepapp);
subplot(1,2,2);
image(Xc);colormap(map);
title('压缩后的图像');
================================
%给出压缩效率
disp('小波分解系数中置 0 的系数个数百分比：');
perf0
disp('压缩后图像剩余能量百分比：');
perfl2
```

结果如图 7-27 所示。

小波分解系数中置 0 的系数个数百分比：

```
perf0 =
   58.0203
```

压缩后图像剩余能量百分比：

```
perfl2 =
    99.9778
```

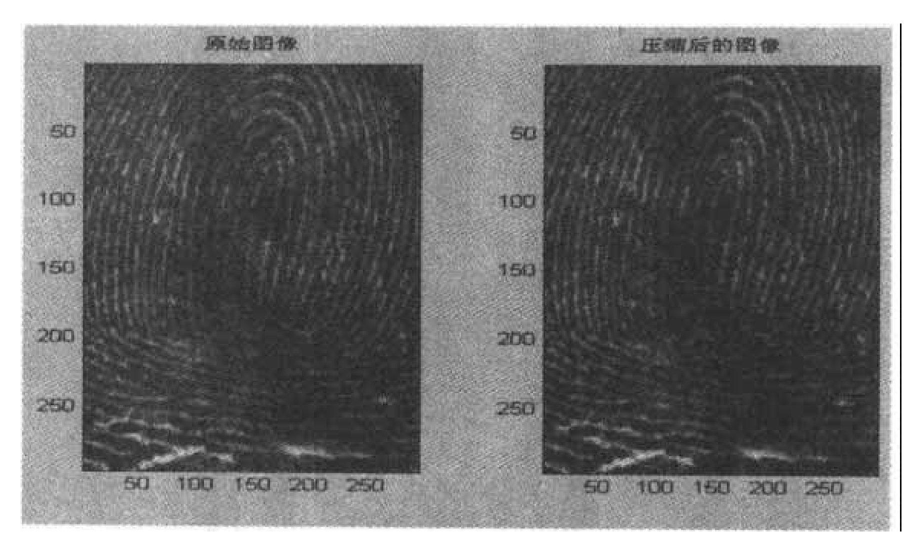


图 7-27　运行结果

从以上的实例中，我们知道，应用小波分解进行图像压缩时，在理论上，可以获得任意压缩比的压缩图像，且实现起来也较为简单。而利用小波包分析进行图像压缩时，随着所选用的分解小波、分解层次和量化阈值的不同所得结果不同，且实现起来也很简单。因

此，小波变换在图像的压缩方面显示出了它的特性：即为图像从空间域变换到时间域提供了一种非常有效的方法，其作用与专门用于图像压缩的离散余弦变换（DCT）、傅里叶变换等类似。应记住一点，任何方法既有区别于其他方法的优点，同时又具有一定的缺陷，因此，要很好地对图像进行压缩处理，需要综合利用多种技术进行。所以利用小波分析进行图像压缩也不例外，往往也是利用小波分析和其他相关技术的有机结合才能达到较为完美的结果。

7.3.3　图像增强

小波变换将一幅图像分解为大小、位置和方向均不相同的分量，在做逆变换之前，可根据需要对不同位置、不同方向上的某些分量改变其系数的大小，从而使得某些感兴趣的分量放大而使某些不需要的分量减小。

下面我们用具体的例子来说明图像的增强。

【例 7-13】用小波分析对所给图像进行增强处理。

例程 7-13

```
%装载并显示原始图像
load flujet;
subplot(1,2,1);
image(X);colormap(map);
title('原始图像');
========================================
%对图像 X 用小波 db3 进行 2 层分解
[c,l]=wavedec2(X,2,'db3');
Csize=size(c);
========================================
%对分解系数做处理以突出所需部分并弱化不需要部分
for i=1:Csize(2)
    if(c(i)>300)
        c(i)=2*c(i);
    else
        c(i)=0.5*c(i);
    end
end
========================================
%重构图像并显示
X1=waverec2(c,l,'db3');
subplot(1,2,2);
image(X1);colormap(map);
title('增强图像');
```

结果如图 7-28 所示。

分解后的图像，其主要信息（即轮廓）由低频部分来表征，而其细节部分则由高频部分表征。因此，在上述的例子中，我们对分解后的低频系数加权进行增强，而对高频部分加权进行弱化，经过如此处理后，即达到了增强图像的目的。

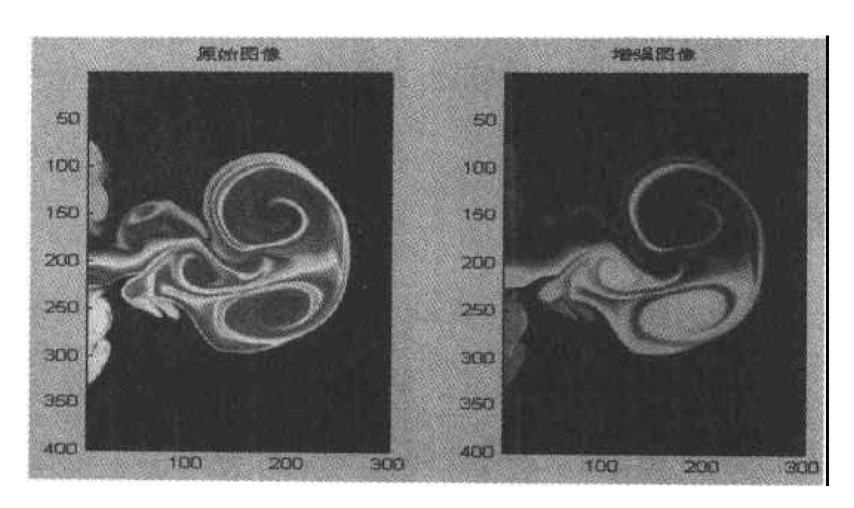


图 7-28　运行结果

7.3.4　图像平滑

我们先看下面的例子。

【例 7-14】 请利用二维小波分析和图像的中值滤波对一给定的含噪图像进行平滑处理。

例程 7-14

```
%装载原始图像
load gatlin;
==========================================
%对图像加噪声并显示出含噪图像
init=2788605800;
randn('seed',init);
X=X+10*randn(size(X));
subplot(1,2,1);
image(X);colormap(map);
title('含噪图像');
==========================================
%图像平滑：应用中值滤波进行处理
for i=2:479
    for j=2:639
        Xtemp=0;
        for m=1:3
            for n=1:3
                Xtemp=Xtemp+X(i+m.2,j+n.2);
```

```
            end
        end
        Xtemp=Xtemp/9;
        X1(i,j)=Xtemp;
    end
end
================================
%显示结果
subplot(1,2,2);
image(X1);colormap(map);
title('平滑图像');
```

结果如图 7-29 所示。

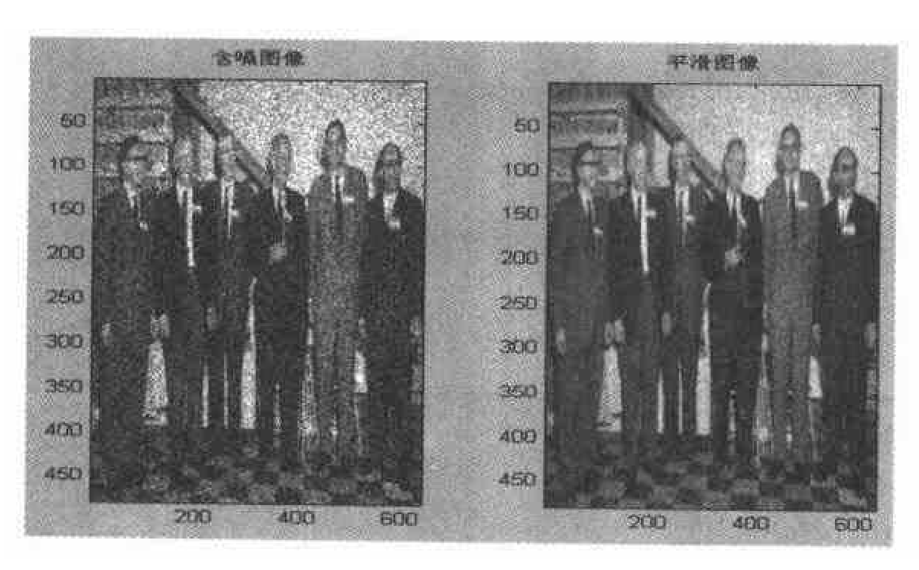


图 7-29　运行结果

在此例中，我们首先对图像加了一个较大的白噪声，之后应用中值滤波对含噪图像进行处理。从图 7-29 中可以看出，经过中值滤波处理后的含噪图像具有较好的平滑效果。

7.3.5　图像融合

图像融合是将同一对象的两个或更多个图像合成在一幅图像之中，以便于满足人们的某种需要。这一技术应用于多频谱图像理解和医学图像处理等领域，使得融合图像更容易为人们所理解。

【例 7-15】请利用二维小波分析将图像 woman.mat 和 wbarb.mat 融合在一起。

例程 7-15

```
%装载原始图像（woman.mat 和 wbarb.mat）并显示
load wbarb;
X1=X;map1=map;
subplot(2,2,1);
```

```
image(X1);colormap(map1);
title('图像 wbarb');
load woman;
X2=X;map2=map;
subplot(2,2,2);
image(X2);colormap(map2);
title('图像 woman');
=====================================
%对上述两图像进行分解
[c1,l1]=wavedec2(X1,2,'sym4');
[c2,l2]=wavedec2(X2,2,'sym4');
%对分解系数进行融合
c=c1+c2;
=====================================
%应用融合系数进行图像重构并显示
XX=waverec2(c,l1,'sym4');
subplot(2,2,3);
image(XX);
title('融合图像 1');
Csize1=size(c1);
=====================================
%对图像进行增强处理
for i=1:Csize1(2)
    c1(i)=1.2*c1(i);
end
Csize2=size(c2);
for j=1:Csize2(2)
    c2(j)=0.8*c2(j);
end
=====================================
%通过减小融合系数以减小图像的亮度
c=0.5*(c1+c2);
=====================================
%对融合系数进行图像重构
XXX=waverec2(c,l2,'sym4');
=====================================
%显示重构结果
subplot(2,2,4);
image(XXX);
title('融合图像 2');
```

结果如图 7-30 所示。

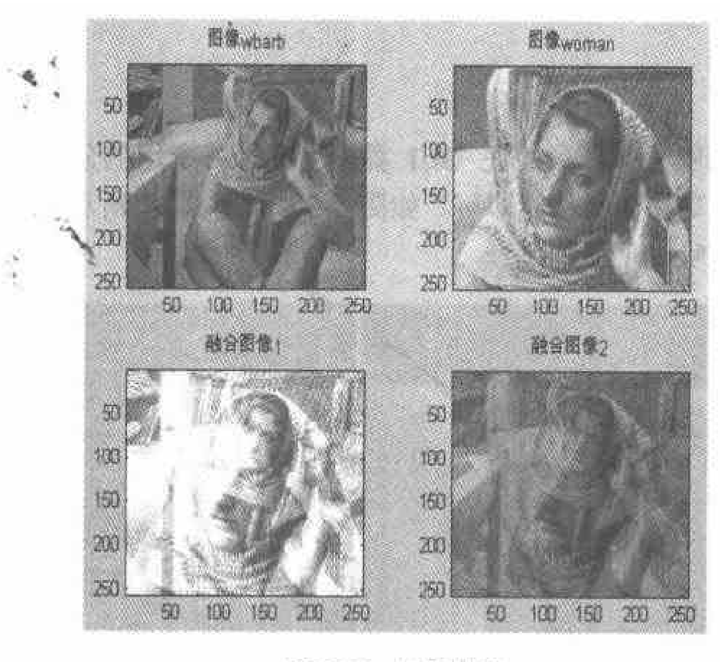


图 7-30 运行结果

7.4 应用 GUI 进行图像处理

在 MATLAB 的小波工具箱中所提供的图形用户界面（GUI），使得在一般目的下的图像处理变得更加容易。下面我们具体介绍如何利用 GUI 进行图像处理。

7.4.1 图像的分解与重构

当打开 GUI，选择了二维小波分析之后，就可装载待处理的图像。如图 7-31 所示。分解图像的过程是应用离散小波变换（DWT）进行的，而重构过程则是逆离散小波变换的应用过程。由图 7-31 的右侧部分可知，我们可以根据不同的需要选择不同的处理方式。

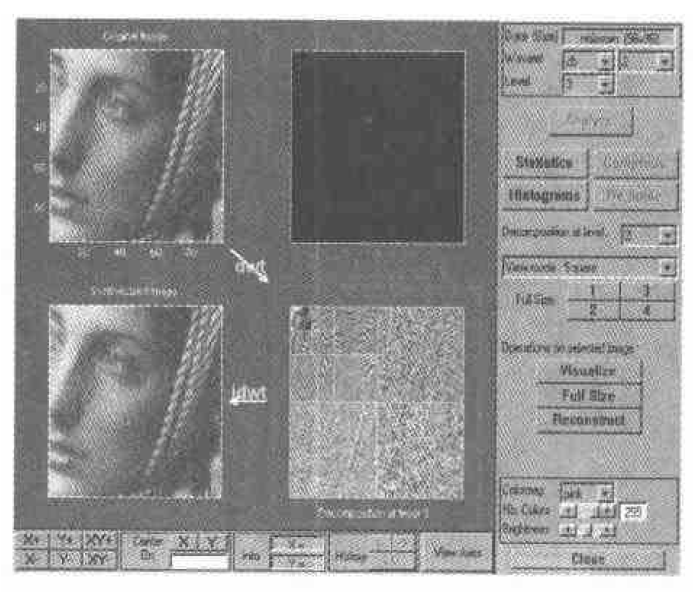


图 7-31 图像的分解与重构

7.4.2 图像消噪

对一幅图像进行消噪处理，只需在 GUI 界面上选择好所用的分析小波、分解层数和所用的量化阈值，然后单击消噪按钮即可，如图 7-32 和图 7-33 所示。下面再看一下应用小波包分析进行图像消噪处理的情况。

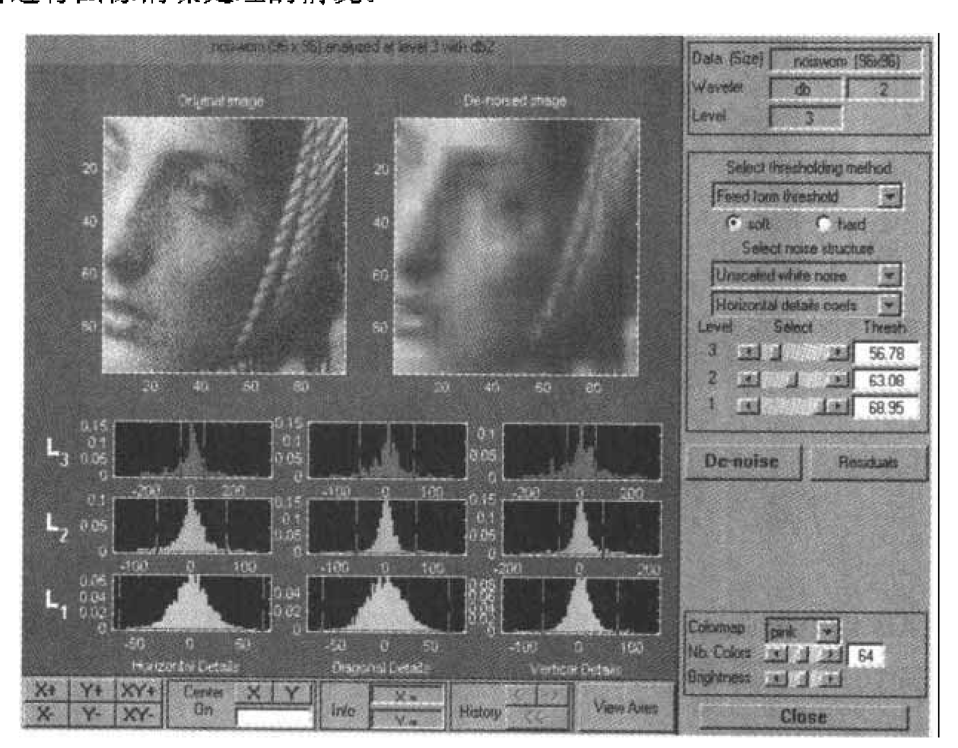


图 7-32　小波分析进行图像消噪处理的结果

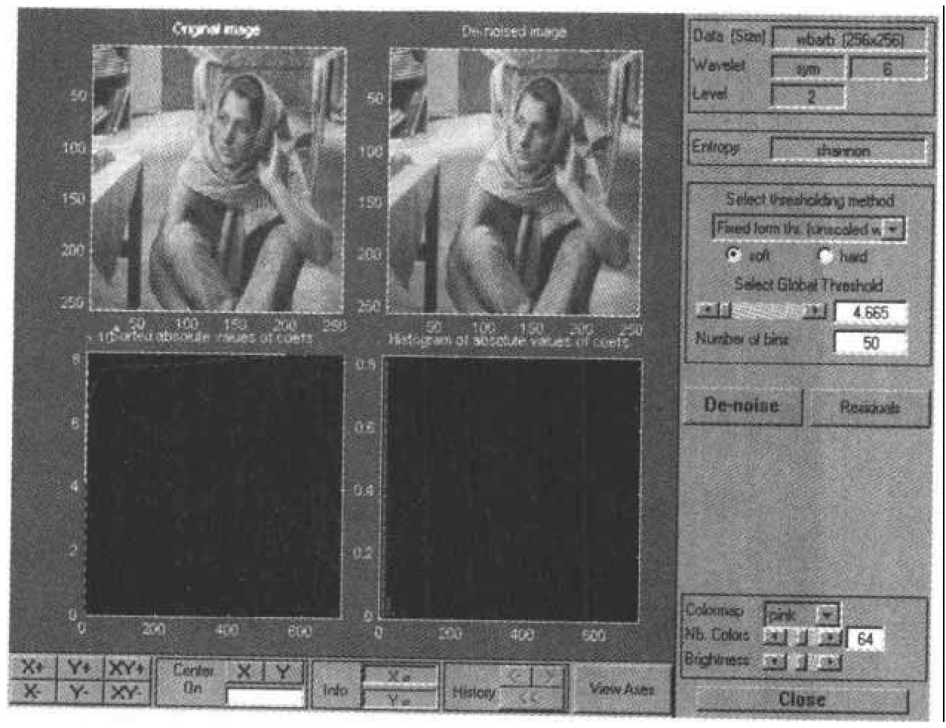


图 7-33　小波包分解图像的结果

选择消噪处理，结果如图 7-34 所示。

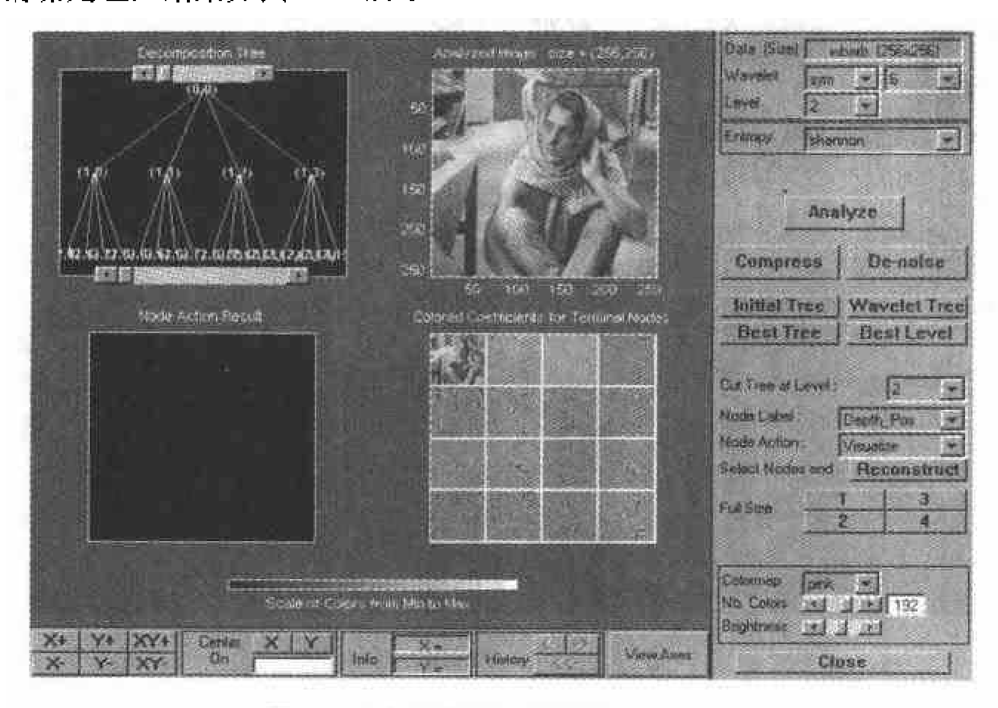


图 7-34　应用小波包分析进行图像消噪

7.4.3　图像压缩

同样地，应用 GUI 进行图像的压缩处理也很方便。当选择好分解小波、分解层数和量化阈值后，即可进行压缩处理。图 7-35 给出了一幅待压缩图像的分解与重构的结果，图 7-36 给出了压缩后的图像，包括压缩后分解系数中置 0 的系数个数百分比、压缩后剩余能量百分比及压缩所用阈值。

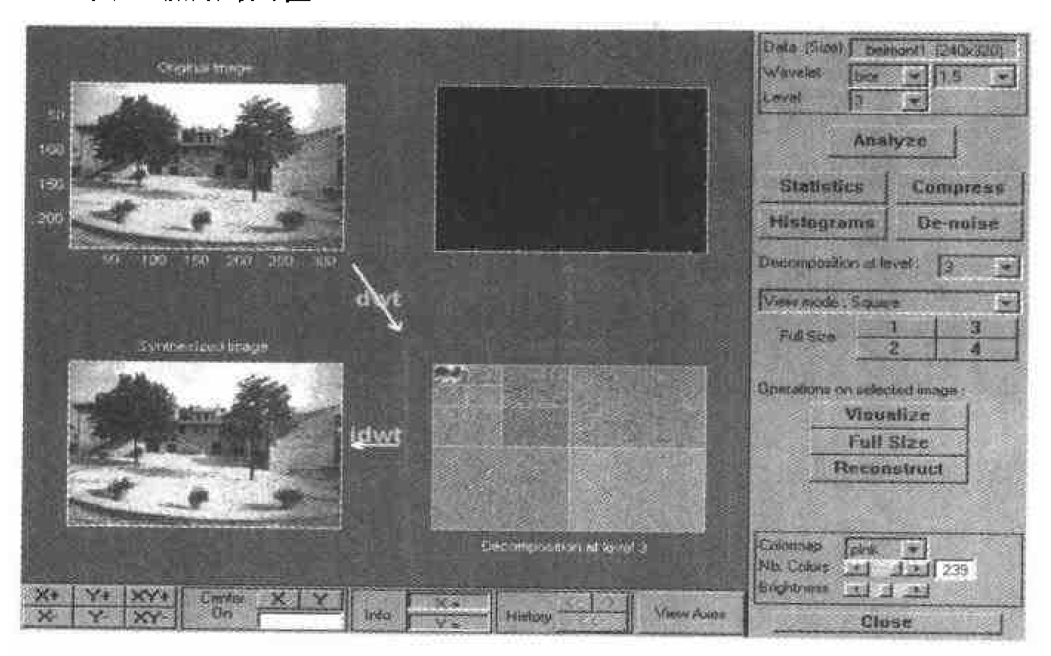


图 7-35　待压缩图像的分解与重构

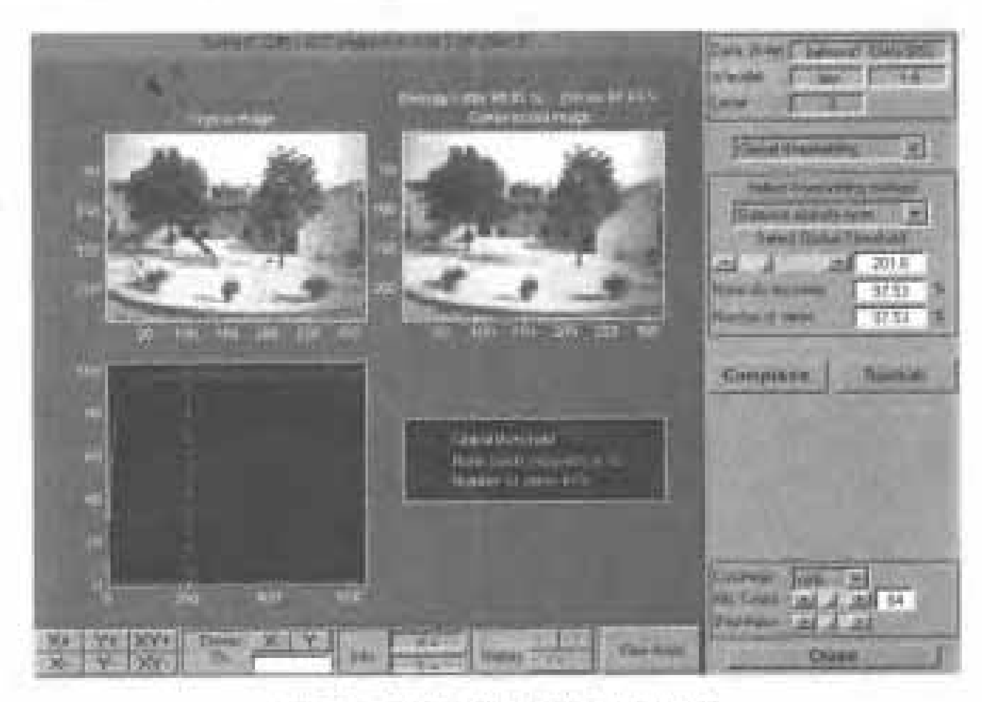


图 7-36　应用小波分析进行图像压缩

在 GUI 中，应用小波包分析进行图像压缩处理的结果如图 7-37 所示。



图 7-37　应用小波包分析进行图像压缩

附录 A MATLAB 6.5 小波工具箱新特性

相对于 MATLAB 5.x 中的小波工具箱 1.0 版，MATLAB 6.5 中的小波工具箱 2.2 版有较大的改进和提高，并表现出很多新的特征。用户可以通过命令 whatsnew wavelet 来查看小波工具箱的这些改进和更新信息。

A.1 新的图形用户接口工具

1. **一维连续小波分析图形工具**（Continuous Wavelet 1-D）

添加了两个新的轴线，即 Coefficients Line（系数线）和 Local Maxima Lines（局部极大值线）。当用户用鼠标在系数的图形表示中选取某个尺度时，Coeffients Lines 就会显示在该尺度上的所有系数组成的曲线。这个功能对于查看连续小波系数很有用。Local Maxima Lines 显示的是系数的局部极大值随尺度变化的曲线，它可以给出一维连续小波变换系数的基本轮廓。另外，在一维连续小波分析图形工具中，只通过鼠标用户就可以在尺度和频率之间很方便地进行转换。

2. **一维复连续小波分析图形工具**（Complex Continuous Wavelet 1-D）

一维复连续小波分析图形工具是采用复小波对实信号进行连续小波分析。对相应小波系数的幅和相位角，显示了三个轴图：

（1）系数（模或相位角）

（2）与用鼠标选择的某个尺度相关的系数线

（3）系数局部极大值线

同样，一维复连续小波分析图形工具也能在尺度和频率之间方便地切换。

一维复连续小波分析图形工具是 MATLAB 6.5 新增的一种小波分析功能，当数据处理过程中需要考虑相位信息时该工具是很有用的。

3. **利用 SWT 1-D 工具进行信号消噪**（Signal De-Noising Using SWT 1-D）

新增的 Stationary Wavelet Transform De-noising 1-D 图形工具用于对含有一些孤立的间断点的染噪信号的去噪，其基本思想是对许多具有微小差别的离散小波消噪过程进行平均。

4. **利用 SWT 2-D 工具进行图像消噪**（Image De-Noising Using SWT 2-D）

新增的 Stationary Wavelet Transform De-noising 2-D 图形工具用于对图像进行消噪，它应用较为简单的消噪策略可以获得较好的消噪效果。

5. **局部自适应阈值处理**（Local Variance Adaptive Thresholding 1-D）

这是小波分析工具箱新增的一项功能。它允许按层次定义随时间变化的阈值，从而增

强消噪策略的灵活性，尤其对于含有非平稳噪声的信号比较有效。小波分析图形工具中具有该项阈值处理功能的模块有DWT 1-D De-noising and Compression、SWT 1-D De-noising和Density and Regression estimation。

6. 密度估计（Density Estimation）

Density Estimation 1-D 也是新增的一项功能，利用小波来进行密度估计，尤其对于那些来自某个未知的不规则密度分布的数据有效，能够克服一些传统的估计方法的缺陷。

7. 回归估计（Regression Estimation）

Regression Estimation 1-D 工具的功能是对确定性和随机性两种模式进行基于小波的回归估计，它适用于对非等间隔采样的染噪信号数据进行消噪或估计两个变量之间的非线性关系，也可以用于数据平滑。

8. **一维小波系数选择工具**（Wavelet Coefficients Selection 1-D）

新增的 Wavelet Coefficients Selection 1-D 工具可以基于下面四种小波系数选取方式进行信号重构：

（1）全局最大值系数（绝对值）选取

（2）分层次最大值系数选取

（3）最大值系数自动选取

（4）手动系数选取

该工具对于手动信号压缩过程中的系数选取很有用。

9. **二维小波系数选择工具**（Wavelet Coefficients Selection 2-D）

新增的 Wavelet Coefficients Selection 2-D 工具可以基于下面三种小波系数选取方式进行图像重构：

（1）全局最大值系数（绝对值）选取

（2）分层次最大值系数选取

（3）最大值系数自动选取

该工具对于手动图像压缩过程中的系数选取很有用。

10. **信号延拓/截断**（Signal Extension/Truncation）

新增的该项功能可以根据需要采用多种方式（periodic，symmetric，smooth和zero-padding）对一维信号进行延拓或截断。例如对信号进行SWT变化时可能就必须进行延拓以满足SWT对信号尺度的要求。

11. **图像延拓/截断**（Image Extension/Truncation）

新增的该项功能可以根据需要采用多种方式（periodic，symmetric，smooth和zero-padding）对二维图像信号进行延拓或截断，例如对图像进行SWT变化时可能就必须进行延拓以满足SWT对信号尺度的要求。

12. 其他剩余信息显示（Residual Display）

新增的该项功能可以在消噪或压缩过程之后显示噪声的频域和时域特性图。小波、小波包、平稳小波系数和阈值处理等图形分析工具里也加入该项功能。

13．动态可视工具（Dynamic Visualization Tool）

这也是新增的一项功能，位于小波分析图形工具箱各个模块窗口底部的【View Axes】按钮就是完成该项功能的。当轴的尺寸较小时（例如在进行压缩或消噪时），该项功能是很有用的。

A.2 新添加的小波函数系列

新增小波函数如表 A-1 所示。

表 A-1 MATLAB 6.5 新增小波函数

小波函数缩写名	简单描述
cmor	Complex Morlet wavelets
cgau	Complex Gaussian wavelets (derivatives of the complex Gaussian function)
fbsp	Complex frequency B-Spline wavelets
shan	Complex Shannon wavelets
sym	Symlets (now available from sym2 to sym45)
rbio	Reverse Biorthogonal wavelets (Reverse Biorthogonal Wavelets based on bior)
dmey	Discrete Meyer wavelet (FIR based approximation of the Meyer Wavelet)
gaus	Gaussian wavelets (derivatives of the Gaussian probability density function)

A.3 新增小波分析工具箱函数

新增小波分析工具箱函数如表 A-2～表 A-6 所示。

表 A-2 平稳小波变变换（Stationary Wavelet Transform）

函数名	功能简述
swt	一维离散平稳小波变换
iswt	一维离散平稳小波逆变换
swt2	二维离散平稳小波变换
iswt2	二维离散平稳小波逆变换

表 A-3　小波消噪和压缩阈值计算（Thresholds for 1-D and 2-D De-noising and Compression）

函　数　名	功 能 描 述
wdcbm	计算一维小波消噪或压缩阈值（采用 Birgé-Massart 算法）
wdcbm2	计算二维小波消噪或压缩阈值（采用 Birgé-Massart 算法）
wbmpen	计算一维或二维小波消噪阈值（基于惩罚策略）
wpbmpen	计算一维或二维小波包消噪阈值（基于惩罚策略）
wthrmngr	阈值管理函数

表 A-4　通用树对象函数（General Tree Object Functions）

函　数　名	功 能 简 述
wptree	构造小波包树（WPTREE）类
dtree	构造类 DTREE（Data Tree）
ntree	构造类 NTREE（N-ary Tree）
get	获取树对象域内容
read	读取树对象域值
set	设置树对象域内容
write	对树对象域赋值
plot	树管理图形工具（该函数产生一个窗口并显示树，在该窗口中可以分解、合并或显示树的每个节点

表 A-5　小波应用函数

函　数　名	功 能 简 述
scal2frq	将尺度转换成频率
centfrq	计算小波中心频率

表 A-6　信号处理函数

函　数　名	功 能 简 述
wvarchg	找出奇异点

A.4　增强功能（Enhancements）

Complex CWT：函数 cwt 新增一项复小波变换功能，即可以采用复小波或实小波对一个实信号进行连续小波变换。

Threshold Settings Using the GUI：消噪和压缩图形工具中的阈值线（threshold lines）可以用鼠标拖动，相应的控制按钮（如编辑框，滑动条等）也会被自动更新。

New Denoising and Compression Methods：图形工具中有了消噪和压缩的新方法，有 Wavelet 1-D, Wavelet 2-D, Wavelet Packet 1-D 和 Wavelet Packet 2-D。

Saving De-noising and Compression Parameters：在大多数图形工具主菜单里，保存选



这些函数在小波分析工具箱 2.2 版中已经不再使用。

新版本中不再使用的函数可以在 MATLAB 6.5 安装目录下的 toolbox/wavelet/waveobsolete 子目录中找到。

命令 wtbxmngr('V2')可以恢复新版本中的小波包数据结构和函数，并且禁止上述 Obsolete Function 的使用。

附录 B　MATLAB 6.5 其他新特性

B.1　MATLAB 6.5 新特性

MATLAB 6.5 包含新的 JIT 加速器(JIT-Accelerator)，对许多运算和数据类型加速器能够显著地提高 MATLAB 的计算速度。其他新特性和功能增强包括：

- 新的 M 文件 Profiler 界面，更好地理解和分析 M 代码；
- 支持 64 位文件句柄，以支持大于 2G 的数据文件的操作；
- 新的 MATLAB Timer 对象，用于规划 MATLAB 命令的执行；
- 增强的自动化客户界面（ActiveX/COM 控件），用于属性浏览和修改新的用户界面，增强的事件和意外处理，支持通过引用方式传递的参数；
- 增强的互联网集成功能，在 MATLAB 下完成读取一个 URL 下的内容，发送 E-mail，压缩和解压缩等；
- 新的开始按钮，方便访问常用的程序；
- 增强的文件和目录管理工具，用于设置或者读取文件和目录的属性，移动和重命名文件和目录；
- 数组编辑器，用于与 Microsoft Excel 之间进行剪切、复制、粘贴、删除，交换单元，支持大数组；
- 数学计算和算法增强：数值积分，微分延迟求解器；
- 许多函数的计算速度更快：Pentium 4 下的满阵和稀疏线性求解器，矩阵乘积，矩阵转置，线性代数等运算；
- 增强的编辑和调试工具：警告信息的灵活控制，新的自动存盘功能；
- 增强的 PC 平台下的源（source control）控制接口：得到最新版本，check in、check out 和删除文件；
- 增强的图形功能：新的 Colormap Editor 和增强的 Property Editor；
- 增强的声音支持：支持 24 位录音，增强的 wav 文件支持；
- 从 HDF 或者 HDF-EOS 导入数据的新的用户图形界面。

B.2　Simulink 5.0 新特性

Simulink 是一种框图模型建模环境，用于动态系统的仿真、性能评估，并进行控制系统、DSP 系统和通信系统的设计。该软件在 GUI 和运行引擎方面的改进有：

- 图形化的程序调试器，使错误分析及调试操作简单起来；

- 支持大多数 Simulink 模块间以矩阵形式交换数据；
- 支持更高速的基于帧信号处理的 DSP 应用；
- 支持大模型开发的新特性，包括更新的工具栏选项和增强的行编辑功能；
- 集成的查找对话框工具，方便了元件在 Simulink 模型和（或）状态流图中的搜索；
- 增加了 Simulink 数据对象，支持用户自定义数据结构；
- 新的航空库及示例；
- 提高了库浏览器和模型浏览器的使用性能；
- 废除了库连接，以方便库模型块的编辑；
- 增强了可配置子系统，便于设计模型向实际实现转化；
- 具有筛选算法的高级查表模块，使仿真和代码生成更加快速、精确、灵活；
- 单一视窗模式节省了宝贵的屏幕空间。

B.3　MATLAB 6.5 新产品

1．航空工具箱（Aerospace Blockset）

航空工具箱以仿真为基础，提供具体的工具用于飞机、飞船、导弹，以及推进系统和子系统的建模、集成和仿真：

- 对于宇航式飞行器，提供了推进、控制系统、系统动力学，以及激励控制箱；
- 用航空工具箱的组件提供了六自由度和三自由度的实例模型；
- 包含重力、空气和风的环境模型；
- 使用空间表示转换模块增强了坐标转换功能；
- 能实现物理特性的单元转换；
- 使用 Handle Graphics 技术，提供了动画模块，实现三自由度和六自由度的动画演示。

2．曲线拟合工具箱（Curve Fitting Toolbox）

曲线拟合工具箱提供了数据预处理的各种方法，比如生成、比较、分析和管理模型。所有的功能既可以以命令行来实现，也可以以图形用户界面来实现：

- 预处理方法，包括数据的标定、截断、滤波和奇异值的去除等；
- 扩展了线性和非线性参数拟合模型库，对于非线性模型，起始点和参数的解算程序都进行了优化；
- 有各种各样的参数和非参数模型，包括最小二乘法、加权最小二乘法和鲁棒拟合过程（包括有参数边界限制和没有参数边界限制）；
- 自定义参数和非参数模型扩展；
- 用样条和内插法实现非参数拟合；
- 拟合数据的插补、外推、微分和积分。

3．Motorola MPC555 嵌入式目标（Embedded Target for Motorola MPC555）

Motorola MPC555 嵌入式目标可以将由 MATLAB 实时工作空间编码器生成的代码直接配置到 MPC555 微控制器。Motorola MPC555 嵌入式目标依靠 MATLAB 实时工作空间编码器来为 Motorola MPC555 生成和提供具体的代码：

- 为 MPC555 提供快速原型和产品代码生成；
- 针对 Motorola MPC555 MIOS、QADC 和 TouCAN 设备驱动块；
- 通过板载调试器(onboard debugger)或 CAN(controller area network)下载应用到 Motorola MPC555；
- 通过处理器在回路联合仿真的方式验证算法代码；
- 生成详细的 HTML 代码生成报告，包括 RAM/ROM 信息。

4．C6000 DSP 平台嵌入式目标（Embedded Target for C6000 DSP Platform）

简化 Texas Instruments DSP 软件设计和分析流程，直接从 MathWorks 环境生成高效代码。您可以使用 Simulink、DSP Blockset 和 Communications Blockset 中的块开发层次化的 DSP 算法框图模型，然后通过 Real-Time Workshop 生成无歧义的可执行算法，并可以进一步由 DSP 软件工程师优化：

- 自动生成支持 Texas Instruments DSP 目标的代码；
- 无需 DSP 编程，就可以实时验证算法；
- 生成 Code Composer Studio 项目格式的带注解的 C 代码。

5．MATLAB COM Builder

MATLAB COM Builder 可以很容易将 MATLAB 算法转化为 COM 对象，可以用于任何基于 COM 的应用：

- 通过简便易用的 GUI 将 MATLAB 算法转换成 COM 对象；
- 生成 COM 对象，可以供 Visual BASIC、C/C++、Microsoft Excel 和其他任何基于 COM 的应用使用；
- 允许修改和查看 MATLAB 生成的代码，可以对您所引用的函数更好地理解；
- 通过 MATLAB 生成的 COM 对象可以自由发布。

6．MATLAB Excel Builder

MATLAB Excel Builder 可以很容易将复杂的 MATLAB 算法转化为独立的 Excel 插件（add-ins），用户可以利用以矩阵为基础、灵活的 MATLAB 编程环境，以及拥有大量可用数学和图像函数的优点，从而迅速建立计算更深入的模型：

- 通过简便易用的 GUI 将 MATLAB 算法转换成 Excel 插件（add-in）；
- 自动生成 .dll 和 Visual BASIC 应用文件，并且可以输入到 Excel 中；
- 生成的插件函数比 Visual BASIC for Applications (VBA)生成的快 95%；
- 允许修改和查看 MATLAB 生成的代码，可以对您所引用的函数背后的逻辑有更好的理解。

7．Code Composer Studio 开发工具的 MATLAB 连接

简化 Texas Instruments DSP 软件设计和分析流程，实现在 TI 软件开发环境实施 DSP

硬件和 MATLAB 之间的通信。算法开发人员、系统设计人员和嵌入工程师通过此工具可以在 MATLAB 中测试、验证和可视化 DSP 软件，消除了 DSP 算法研究和实现之间的缺口：

- 提供 M 文件功能，允许您对数据传输和验证任务进行定制和自动化；
- 在 MATLAB 和 Texas Instruments DSP 设备之间提供数据传输能力，而不必停止目标应用上的执行的任务；
- 硬件对象和 Code Composer Studio 帮助测试，验证和可视化 DSP 代码。

8. 基于模型校准工具箱（Model-Based Calibration Toolbox）

基于模型校准工具箱提供设计工具来校准功率系统。它是建立在 MATLAB 高性能的计算环境及仿真能力的基础上，减少了功率器的检测时间，提高了工程效率，节省了校准时间，因而有潜力改进功率系统性能和可靠性：

- 新的经典设计：Plackett-Burman 两水平筛选设计（tow-level screening design），规则单纯形（Regular Simplex），一个效率高的一阶设计方法；
- 使用更方便，可以将详细用户信息存储起来，可以在 Model Browser 中跟踪项目文件的历史列表；
- 新的上下文菜单，可以复制、删除和重命名模型；
- 在现有的所有模型绘图放大功能之上，增加了新的数据绘图放大工具；
- 在 CAGE 用户界面中浏览大型数据集，速度可以提高 40%以上；
- 支持 Simulink 库中的高级查表类型。

9. 机械仿真（SimMechanics）

它允许工程师在 Simulink 环境中仿真机械系统。应用 SimMechanics 可以直接建立机械部件的模型，仿真它们的运动及分析结果，而不需要推导数学方程：

- 增强的非线性方程求解器，仿真速度更快；
- 优化的模型线性化；
- 新的邻接（adjoining）特性，方便对相邻的其他物体的坐标系位置和方向的定义；
- 对于并联机械模型，用户可以选择和观察切断铰链（cut joints）；
- 图形用户界面增强，模型动化功能增强。

具体内容读者可到网站 http://www.mathworks.com 查阅。

B.4 MATLAB 6.5 主要更新的产品

1. Communications Blockset 2.5

提供了设计和仿真通信系统物理层的模块库，新的特性包括：

- 新的 RF Impairments 库支持考虑非线性，相位噪声，热噪声，路径衰减，I/Q 不均衡，频率漂移，相位漂移；
- 新的序列生成器，生成正交和伪随机序列用于传输和同步；
- 新的循环冗余校验（CRC）库支持 CRC 生成和出错检查；

- 增强的输出功能，显示眼图、散点图和信号轨迹；
- 增强的 RS 编码/解码块，提供更多可利用的编码和快速仿真；
- 新的应用实例，完整的端对端通信链路，包括 WCDMA, HiperLAN/2 和 Bluetooth。

2. Control System Toolbox 5.2

提供了一个对反馈控制系统交互建模、分析和设计环境。您可以对连续和离散线性动态系统进行分析和仿真。新的特性包括：

- SISO 模型可以直接接受 System Identification Toolbox 的输入，不再需要在命令行的转换；
- 增强的系统响应绘图，包括 LTI viewer 中实时更新显示；
- 两个新的反馈结构，在内环上前馈和极联滤波器；
- 补偿器设计存档 GUI 允许方便地对多个补偿器管理。

3. DSP Blockset 5

包括 Simulink 库用来设计和仿真 DSP 应用。这些库包含经典的、多速率的和自适应滤波、转换、矩阵运算，以及线性代数、统计和谱分析。主要的增强包括：

- 所有块全面支持单精度；
- 全面支持嵌入实时目标 (ERT) 生成的 ANSI C 代码(需要 Real-Time Workshop Embedded Coder)；
- 大多数块支持定点数据类型；
- 产生的代码效率更高；
- 新的 Digital Filter 块，实现预先定义的数字滤波器。

4. Fixed-Point Blockset 4

仿真定点对系统的影响，研究字长、量化和定标等因素的影响。主要的功能增强包括：

- 将定点和内置块合二为一；
- 安装和 license 改变；
- 增强的数据类型支持。

5. MATLAB Compiler 3

对 MATLAB 的扩展，自动将 MATLAB 应用转换为独立的 C 和 C++ 代码。

MATLAB C/C++ Math 和 Graphics Libraries 已经和并到 MATLAB Compiler 3 成为单独的产品。这些库不再作为独立的产品支持。

对于现有的 MATLAB Compiler 或者 MATLAB C/C++ Math 和 Graphics Libraries 用户，关于如何得到此产品的详细信息参见 MathWorks Release 13 重要产品变更。

6. Real-Time Workshop 5

Simulink 的补充，能够自动从 Simulink 模型生成代码。新的特性包括：

- 改进的 Real-Time Workshop/Stateflow 接口；
- 修订的生成代码文件打包（Packaging）；
- 表达式折叠（Expression folding）增强及 API 文档；
- 增加 ERT 的外部模式的支持；



- 非虚（Non-virtual）系统的代码重利用；
- Real-Time Workshop 与 Stateflow 代码生成融为一体；
- 生成 HTML 报告选项同样适用其他生成对象。

7．Real-Time Workshop Embedded Coder 3

Real-Time Workshop 功能扩展，为 Simulnk 模型提供高质量嵌入代码生成，从 Stateflow 中(使用 Stateflow Coder)生成产品代码。在 R13 中有显著的增强：

- 支持 Simulink 外部模式；
- 可以生成可重复利用的代码；
- 针对浮点数学环境对象；
- 交互学习教材；
- 新的用户使用手册。

8．Signal Processing Toolbox 6

提供了利用对象的方式建立、观察、修改离散时间滤波器和加窗的新方法，新的特性和增强包括：

- 新的面向目标数字滤波对象和窗口对象；
- 加窗设计和分析工具，用于设计和比较加窗；
- 加窗可视化工具用来显示加窗；
- 用于加窗、滤波器和滤波器分析的新函数；
- 滤波器设计、分析工具（FDATool）和滤波器可视化工具（FVTool）增强。

9．Stateflow 5

对复杂的事件驱动系统进行建模与仿真的交互设计环境。基于有限状态机理论，并且与 MATLAB 和 Simulink 无缝集成。Stateflow 提供了一个最佳的方法来设计嵌入系统的监督逻辑。新的特性包括：

- 对定点数据和运算的全方位支持；
- Stateflow Editor 支持 undo 和 redo；
- 完整的 API 文档，可以通过编程方式使用 Stateflow；
- 支持二维矩阵，输入和输出 Simulink 矩阵信号；
- Stateflow Debugger 现在支持数据溢出检测；
- 通过输出库状态图中的图形函数，可以支持代码重利用。

10．Stateflow Coder 5

从 Stateflow 状态图中生成高质量代码，增强包括：

- 与 Real-time Workshop 紧密集成；
- 可以指定在生成代码中内嵌图形函数，最大限度提高代码效率；
- 图形函数中的不必要的初始化部分被删除；
- 优化简单的 if 语句；
- 在代码中对 for 循环的使用优化；
- 用户在 Stateflow 状态图中输入的注释包含在生成的代码中。

11．Statistics Toolbox 4

提供新的多变量技术，增强的试验设计功能和概率分布及新的函数支持：

- 用于分类和回归分析中决策树（Decision Tree）法；
- 因素分析（Factor analysis）；
- K-means 聚类方法，典型相关分析（canonical correlation）、多维等级分析（multidimensional scaling）和 Procrustes rotation；
- 从 Wishart 和逆 Wishart 分布生成随机矩阵，或进行拉丁超立方体抽样（latin hypercube sampling）；
- Central composite 和 Box-Behnken 试验设计法；
- 利用 kernel smoothing 法估计概率密度和为调查数据计算经验累计分布函数中；
- 扩展了分类，D-optimal 设计和概率分布的功能。

12．Virtual Reality Toolbox 3

三维动画显示 Simulink 系统，提供以下新特性：

- 跨平台支持（PC 和 UNIX）；
- Virtual Reality Toolbox 浏览器，提供一个选项，即可以将 VR 客户端作为 Web 浏览器的一个插件，也可以作为 Virtual Reality Toolbox 浏览器；
- 块描述（Block description）中的工具箱演示帮助链接；
- 增加了点标记，处理 VR 对象。

B.5　MATLAB 6.5 其他更新的产品

关于下列产品的更新信息，请参见 Release Notes ：

- Communications Toolbox 2.1
- Data Acquisition Toolbox 2.2
- Excel Link 2
- Filter Design Toolbox 2.2
- Financial Derivatives Toolbox 2
- Financial Time Series Toolbox 2
- Image Processing Toolbox 3.2?/td>
- Instrument Control Toolbox 1.2
- Instrument Control Toolbox 1.2
- Mapping Toolbox 1.3
- MATLAB Report Generator1.3
- Optimization Toolbox 2.2
- Real-Time Windows Target 2.2
- SimPower Systems 2.3 (formerly named Power System Blockset)
  对于现有的 Power System Blockset 用户，关于如何得到此产品的详细信息参见



MathWorks Release 13 重要产品变更。

- Simulink Performance Tools 1.2
- Simulink Report Generator 1.3
- Wavelet Toolbox 2.2
- xPC Target 2
- xPC Target Embedded Option 2

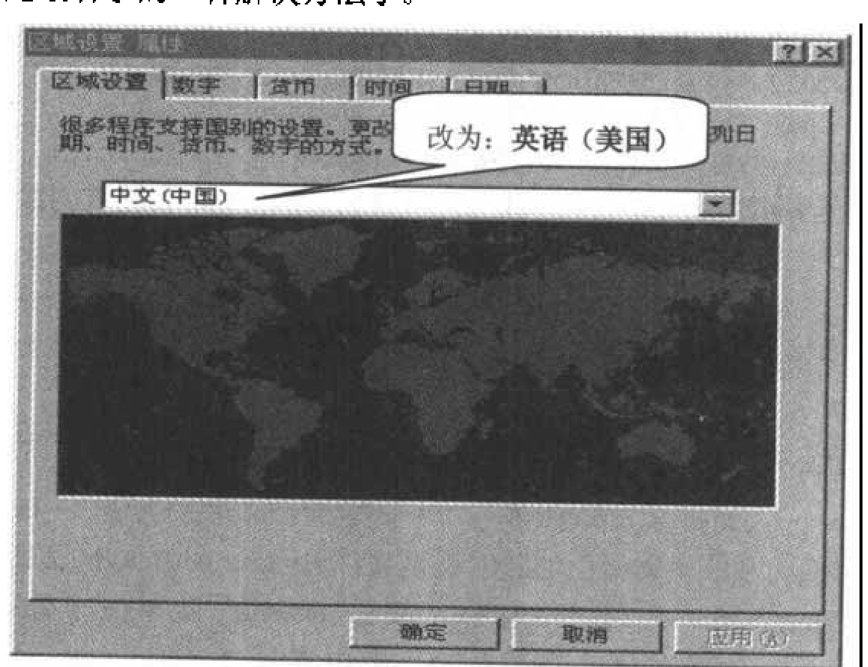
附录C　MATLAB 6.5 安装问题指南

C.1　MATLAB 6.5 为什么安装上不能启动

“MATLAB 6.5 为什么安装上不能启动？”，经常有朋友遇到这个问题，其实在Windows 98或其第2版下安装MATLAB 6.5应该没有问题(也有部分可能会在安装了IE 5.5或更高版本，以及更新了Windows的一些字库以后出现)，而在Windows Me或Windows 2000下安装MATLAB 6.5不能启动的主要原因是它的部分中文字库和操作系统的字库重名造成冲突，下面给出几种可行的解决方法以供选择。

1．**更改区域设置**

在控制面板里有区域设置一项，一般中文的Windows操作系统上默认的区域设置是“中文（中国）”，如图C-1所示。这里我们只要把它更改为“**英语（美国）**”之后重新启动计算机，MATLAB 6.5就可以正常地运行了。不用担心，改过区域设置之后，操作系统仍然是中文操作系统，并不会给我们的使用带来太多的麻烦（当然会有些小的显示等方面的问题）。这也是最省事的一种解决方法了。



图C-1　更改区域设置

2．**消除重名字库**

可以直接在操作系统中将重名字库给删除，或是在MATLAB 6.5的字体配置文件中改变重名字库相关信息。前一种删除方法将导致该字库从操作系统中被完全删掉，其他应用

软件也将无法使用该字库，因此并不十分理想，不过却也非常方便。后一种方法，该字库没有从操作系统中删掉，只是在 MATLAB 6.5 的字体属性配置文件中改变了相关信息，因此并不影响其他应用软件使用这些字库。这两种删除方法共同的最核心的问题就在于要知道是哪一些字库重名了，而这往往需要去试验或是获得他人的经验。目前在一般的情况下，大约只有"新宋体"这个字库有重名的问题。第一种删除字体的方法也非常简单（在字体目录下将相应的文件删除即可），但一般不提倡使用。这里，我们就介绍如何更改 MATLAB 6.5 的字体配置文件。

首先，MATLAB 6.5 这个导致死机的字体属性配置文件是 font.properties.zh，这个文件在目录$MATLAB\sys\java\jre\win32\jre\lib\下（$MATLAB 表示 MATLAB 6.5 的安装目录），我们可以用一个文本编辑器来编辑这个文件。

在 $MATLAB\sys\java\jre\win32\jre\lib\font.properties.zh 文件的最后加上：

```
# Font entry added by The MathWorks:"新宋体"
tmw36525210.0=\u65b0\u5b8b\u4f53,ANSI_CHARSET
tmw36525210.1=\u65b0\u5b8b\u4f53,GB2312_CHARSET
tmw36525210.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
tmw36525210.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED
fontcharset.tmw36525210.2=sun.awt.windows.CharToByteWingDings
fontcharset.tmw36525210.3=sun.awt.CharToByteSymbol

# Font entry added by The MathWorks:"@新宋体"
tmw39767002.0=@\u65b0\u5b8b\u4f53,ANSI_CHARSET
tmw39767002.1=@\u65b0\u5b8b\u4f53,GB2312_CHARSET
tmw39767002.2=WingDings,SYMBOL_CHARSET,NEED_CONVERTED
tmw39767002.3=Symbol,SYMBOL_CHARSET,NEED_CONVERTED
fontcharset.tmw39767002.2=sun.awt.windows.CharToByteWingDings
fontcharset.tmw39767002.3=sun.awt.CharToByteSymbol
```

以上结果为在 win2k sc pro 上所得，如果您的系统中还有其他有问题的中文字库，需要同样的处理。经过检验可以找到一些没有问题的中文字库如：宋体、黑体、仿宋、楷体、幼圆和隶书。

这种方法的两种手段都比较麻烦，需要一些经验和试验，如果不小心还可能会出一些问题，下面我们就介绍 Mathworks 公司的解决方案。

3．使用 Mathworks 公司提供的技术支援

经过 Mathworks 公司技术人员的研究，终于找到了问题的根源——Java 虚拟机用来将标准 Java 字体转换为系统字体所用的映射属性文件 font.properties.zh 及其映射程序中存在一个小 BUG。Mathworks 公司已经将补丁文件 mwt.jar 发布在其公司的网站 http://www.mathworks.com 上了，该文件最新的下载位置是 ftp://ftp.mathworks.com/pub/tech-support/solutions/s26990。下载了该文件后将其替换目录$MATLAB\java\jar\下的同名文件（当然最好对初始的文件做一个备份）。然后，重新启动 MATLAB 6.5，问题就可以解决了，如果仍有一些其他问题，可以到其公司或其他一些相关网站上查询，也可以发邮件给作者，欢迎互相交流。

C.2　安装时的更新 Java 虚拟机的问题

安装时的更新 Java 虚拟机的问题一般比较偶而地出现在 Windows 98 和 Windows NT 环境中，有的时候因为安装其他软件时已经解决了这个问题，在安装 MATLAB 6.5 时就会比较顺利。但偶而确实会出现要求更新 Java 虚拟机的提示消息框，这时我们应当更新我们的 Java 环境。其软件一般可以在 MATLAB 6.5 的安装盘中找到，如果是下载的软件，可以再到网址 http://www.microsoft.com/java/vm/dl_vm40.htm 上去下载相应的软件 msvm.exe（对应于 Windows 98 和 Windows NT 操作系统）。如果是 Windows 2000 操作系统，只需要安装升级补丁 SP2。

C.3　PDF 文档的获取

国内的 MATLAB 6.5 软件大多是下载的软件，因此帮助文件不全。下面的这两个网址可以给大家一点帮助：

http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml

http://science.fire.ustc.edu.cn/matlab6.html

附录D　MATLAB 6.5 小波分析工具箱函数

D.1　一 般 函 数

1．biorfilt

功能：产生双正交小波滤波器组

语法：

```
[Lo_D,Hi_D,Lo_R,Hi_R] = biorfilt(DF,RF)
[Lo_D1,Hi_D1,Lo_R1,Hi_R1,Lo_D2,Hi_D2,Lo_R2,Hi_R2] =
biorfilt(DF,RF,'8')
```

举例：

```
[Rf,Df] = biorwavf('bior3.5');
[Lo_D,Hi_D,Lo_R,Hi_R] = biorfilt(Df,Rf);
```

参考：biorwavf, orthfilt

2．centfrq

功能：计算小波中心频率

语法：

```
FREQ = centfrq('wname')
FREQ = centfrq('wname',ITER)
[FREQ,XVAL,RECFREQ] = centfrq('wname',ITER,'plot')
```

举例：

```
wname = 'db2';
iter = 8;
cfreq = centfrq(wname,8,'plot')

cfreq =
    0.6667
```

参考：无

3．dyaddown

功能：信号的二倍抽取

语法：

```
Y = dyaddown(X,EVENODD)
Y = dyaddown(X)
```

```
Y = dyaddown(X,EVENODD,'type')
Y = dyaddown(X,'type',EVENODD)
```

举例：

```
s = 1:10
s =
        1   2   3   4   5   6   7   8   9   10

dse = dyaddown(s)    % Downsample elements with even indices.
dse =
        2   4   6   8   10
dse = dyaddown(s,0)
dse =
        2   4   6   8   10

dso = dyaddown(s,1) % Downsample elements with odd indices.
dso =
        1   3   5   7   9
```

参考： dyadup

4、dyadup

功能： 信号的二倍补零插值

语法：

```
Y = dyadup(X,EVENODD)
Y = dyadup(X)
Y = dyadup(X,EVENODD,'type')
Y = dyadup(X,'type',EVENODD)
```

举例：

```
% For a vector.
s = 1:5
s =
    1 2 3 4 5

dse = dyadup(s) % Upsample elements at odd indices.
dse =
    0 1 0 2 0 3 0 4 0 5 0
```

参考： dyaddown

5、intwave

功能： 计算小波的积分函数

语法：

```
[INTEG,XVAL] = intwave('wname',PREC)
[INTEG,XVAL] = intwave('wname',PREC,PFLAG)
[INTEG,XVAL] = intwave('wname')
```

举例：

```
wname = 'db4';
```

```
[phi,psi,xval] = wavefun(wname,7);
subplot(211); plot(xval,psi); title('Wavelet');
[integ,xval] = intwave(wname,7);
subplot(212); plot(xval,integ);
title(['Wavelet integrals over [-Inf x] ' ...
       'for each value of xval']);
```

参考： wavefun

6. orthfilt

功能： 产生正交小波滤波器组

语法：

```
[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt(W)
```

举例：

```
load db8; w = db8;
subplot(421); stem(w);
title('Original scaling filter');
[Lo_D,Hi_D,Lo_R,Hi_R] = orthfilt(w);
subplot(423); stem(Lo_D);
title('Decomposition low-pass filter');
subplot(424); stem(Hi_D);
title('Decomposition high-pass filter');
subplot(425); stem(Lo_R);
title('Reconstruction low-pass filter');
subplot(426); stem(Hi_R);
title('Reconstruction high-pass filter');
df = [Lo_D;Hi_D];
rf = [Lo_R;Hi_R];
id = df*df'
id =
     1.0000         0
          0    1.0000
id = rf*rf'
id =
     1.0000         0
          0    1.0000
df = [Lo_D 0 0;Hi_D 0 0];
dft = [0 0 Lo_D; 0 0 Hi_D];
zer = df*dft'
zer =
     1.0e-12 *
     -0.1883 0.0000
     -0.0000 -0.1883
fftld = fft(Lo_D); ffthd = fft(Hi_D);
freq = [1:length(Lo_D)]/length(Lo_D);
subplot(427); plot(freq,abs(fftld));
title('Transfer modulus: low-pass');
```

```
subplot(428); plot(freq,abs(ffthd));
title('Transfer modulus: high-pass')
```

参考：biorfilt, qmf, wfilters

7．truesize

功能：构造二次镜像滤波器组

语法：

```
Y = qmf(X,P)
Y = qmf(X)
```

举例：

```
load db10;
subplot(321); stem(db10); title('db10 low-pass filter');
qmfdb10 = qmf(db10);
subplot(322): stem(qmfdb10); title('QMF db10 filter');
m = fft(db10);
mt = fft(qmfdb10);
freq = [1:length(db10)]/length(db10);
subplot(323); plot(freq,abs(m));
title('Transfer modulus of db10')
subplot(324); plot(freq,abs(mt));
title('Transfer modulus of QMF db10')
subplot(325): plot(freq,abs(m).^2 + abs(mt).^2);
title('Check QMF condition for db10 and QMF db10')
xlabel(' abs(fft(db10))^2 + abs(fft(qmf(db10))^2 = 1')
```

参考：无

8．scal2frq

功能：尺度到频率的变换

语法：

```
F = scal2frq(A,'wname',DELTA)
```

举例：

```
delta = 0.1; wname = 'coif3';
amax = 7; a = 2.^[1:amax];
f = scal2frq(a,wname,delta);
per = 1./f;
```

参考：centfrq

9．wavefun

功能：产生小波函数和尺度函数

语法：

```
[PHI,PSI,XVAL] = wavefun('wname',ITER)
[PHI1,PSI1,PHI2,PSI2,XVAL] = wavefun('wname',ITER)
[PSI,XVAL] = wavefun('wname',ITER)
```

```
[...] = wavefun('wname',A,B)
```

举例：

```
iter = 10;
wav = 'sym4';
for i = 1:iter
    [phi,psi,xval] = wavefun(wav,i);
    plot(xval,psi);
    hold on
end
title(['Approximations of the wavelet ',wav, ...
        ' for 1 to ',num2str(iter),' iterations']);
hold off
```

参考：intwave, waveinfo, wfilters

10. wavefun2

功能：产生二维小波函数和尺度函数

语法：

```
[S,W1,W2,W3,XYVAL] = wavefun2('wname',ITER)
[S,W1,W2,W3,XYVAL] = wavefun2('wname',ITER,'plot')
[S,W1,W2,W3,XYVAL] = wavefun2('wname',A,B)
```

举例：

```
iter = 4;
wav = 'sym4';
[s,w1,w2,w3,xyval] = wavefun2(wav,iter,0);
```

参考：intwave, wavefun, waveinfo, wfilters

11. wavemngr

功能：小波管理函数

语法：

```
wavemngr('add',FN,FSN,WT,NUMS,FILE)
wavemngr('add',FN,FSN,WT,NUMS,FILE,B)
wavemngr('del',N)
wavemngr('restore')
wavemngr('restore',IN2)
OUT1 = wavemngr('read')
OUT1 = wavemngr('read',IN2)
OUT1 = wavemngr('read_asc')
```

举例：

```
wavemngr('add','Lemarie','lem',1,'1 2 3 4 5','lemwavf');
```

参考：无

12. wfilters

功能：由正交或双正交的小波构造滤波器组

语法：

```
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters('wname')
```

```
[F1,F2] = wfilters('wname','type')
```

举例：

```
wname = 'db5';
[Lo_D,Hi_D,Lo_R,Hi_R] = wfilters(wname);
subplot(221); stem(Lo_D);
title('Decomposition low-pass filter');
subplot(222); stem(Hi_D);
title('Decomposition high-pass filter');
subplot(223); stem(Lo_R);
title('Reconstruction low-pass filter');
subplot(224); stem(Hi_R);
title('Reconstruction high-pass filter');
xlabel('The four filters for db5')
```

参考：biorfilt, orthfilt, waveinfo

13．wmaxlev

功能：给出信号在给定小波情况下的最大分解层数

语法：

```
L = wmaxlev(S,'wname')
```

举例：

```
s = 2^10;
w = 'db1';
l = wmaxlev(s,w)
l =
     10
w = 'db7';
l = wmaxlev(s,w)
l =
     6
```

参考：wavedec, wavedec2, wpdec, wpdec2

D.2　小波函数族

1．biorwavf

功能：双正交样条小波滤波器

语法：

```
[RF,DF] = biorwavf(W)
```

举例：

```
wname = 'bior2.2';
[rf,rd] = biorwavf(wname)
rf =
    0.2500    0.5000    0.2500
df =
```

```
-0.1250    0.2500    0.7500    0.2500 -0.1250
```

参考：biorfilt, waveinfo

2．cgauwavf

功能：复数高斯小波

语法：

```
[PSI,X] = cgauwavf(LB,UB,N,P)
```

举例：

```
lb = -5; ub = 5; n = 1000;
[psi,x] = cgauwavf(lb,ub,n,4);
subplot(211)
plot(x,real(psi)),
title('Complex Gaussian wavelet of order 4')
xlabel('Real part'), grid
 subplot(212)
 plot(x,imag(psi))
 xlabel('Imaginary part'), grid
```

参考：waveinfo

3．cmorwavf

功能：复数 Morlet 小波

语法：

```
[PSI,X] = cmorwavf(LB,UB,N,FB,FC)
```

举例：

```
fb = 1.5; fc = 1;
lb = -8; ub = 8; n = 1000;
[psi,x] = cmorwavf(lb,ub,n,fb,fc);
subplot(211)
plot(x,real(psi)),
title('Complex Morlet wavelet cmor1.5-1')
xlabel('Real part'), grid
subplot(212)
plot(x,imag(psi))
xlabel('Imaginary part'), grid
```

参考：waveinfo

4．coifwavf

功能：Coiflet 小波滤波器

语法：

```
F = coifwavf(W)
```

举例：

```
wname = 'coif2';
f = coifwavf(wname)
f =
Columns 1 through 7
```

```
0.0116   -0.0293   -0.0476 0.2730 0.5747   0.2949   -0.0541
Columns 8 through 12
-0.0420   0.0167   0.0040 -0.0013 -0.0005
```

参考：waveinfo

5. dbaux

功能：计算 Daubechies 小波滤波器

语法：

```
W = dbaux(N,SUMW)
W = dbaux(N)
```

举例：

```
P = [ -1/16 0 9/16 1 9/16 0 -1/16]
P =
   -0.0625         0   0.5625   1.0000   0.5625         0 -0.0625
rP = roots(P);
ww = poly([rP(6) -1 -1]);      % filter construction
ww = ww / sum(ww)                  % normalize sum
ww =
    0.3415   0.5915   0.1585 -0.0915
w = dbaux(2)
w =
    0.3415   0.5915   0.1585 -0.0915
```

参考：dbwavf, wfilters

6. dbwavf

功能：Daubechies 小波滤波器

语法：

```
F = dbwavf(W)
```

举例：

```
wname = 'db4';
f = dbwavf(wname)
f =
Columns 1 through 7
0.1629   0.5055   0.4461 -0.0198 -0.1323   0.0218   0.0233
Column 8
-0.0075
```

参考：dbaux, waveinfo, wfilters

7. fbspwavf

功能：复频率 B 样条小波滤波器

语法：

```
[PSI,X] = fbspwavf(LB,UB,N,M,FB,FC)
```

举例：

```
m = 2; fb = 1; fc = 0.5;
```

```
lb = -20; ub = 20; n = 1000;
[psi,x] = fbspwavf(lb,ub,n,m,fb,fc);
subplot(211)
plot(x,real(psi))
title('Complex Frequency B-Spline wavelet fbsp2-1-0.5')
xlabel('Real part'), grid
subplot(212)
plot(x,imag(psi))
xlabel('Imaginary part'), grid
```

参考： waveinfo

8．gauswavf

功能： 高斯小波

语法：

```
[PSI,X] = gauswavf(LB,UB,N,P)
```

举例：

```
lb = -5; ub = 5; n = 1000;
[psi,x] = gauswavf(lb,ub,n,8);
plot(x,psi),
title('Gaussian wavelet of order 8'), grid
```

参考： waveinfo

9．mexihat

功能： 墨西哥草帽小波

语法：

```
[PSI,X] = mexihat(LB,UB,N)
```

举例：

```
lb = -5; ub = 5; n = 1000;
[psi,x] = mexihat(lb,ub,n);
plot(x,psi), title('Mexican hat wavelet')
```

参考： waveinfo

10．meyer

功能： Meyer 小波

语法：

```
[PHI,PSI,T] = meyer(LB,UB,N)
[PHI,T] = meyer(LB,UB,N,'phi')
[PSI,T] = meyer(LB,UB,N,'psi')
```

举例：

```
lb = -8; ub = 8; n = 1024;
[phi,psi,x] = meyer(lb,ub,n);
subplot(211), plot(x,psi)
title('Meyer wavelet')
subplot(212), plot(x,phi)
title('Meyer scaling function')
```

参考：meyeraux, wavefun, waveinfo

11．meyeraux

功能：Meyer 小波辅助函数

语法：

```
Y = meyeraux(X)
```

举例：

无

参考：meyer

12．morlet

功能：Morlet 小波

语法：

```
[PSI,X] = morlet(LB,UB,N)
```

举例：

```
lb = -4; ub = 4; n = 1000;
[psi,x] = morlet(lb,ub,n);
plot(x,psi), title('Morlet wavelet')
```

参考：waveinfo

13．rbiowavf

功能：逆双正交样条小波滤波器

语法：

```
[RF,DF]= rbiowavf(W)
```

举例：

```
wname = 'rbio2.2';
[rf,df] = rbiowavf(wname)
rf =
    -0.1250    0.2500    0.7500    0.2500   -0.1250
df =
     0.2500    0.5000    0.2500)
```

参考：biorfilt, waveinfo

14．shanwavf

功能：复数 Shannon 小波

语法：

```
[PSI,X] = shanwavf(LB,UB,N,FB,FC)
```

举例：

```
fb = 1; fc = 1.5;
lb = -20; ub = 20; n = 1000;
[psi,x] = shanwavf(lb,ub,n,fb,fc);
subplot(211)
plot(x,real(psi)),
title('Complex Shannon wavelet shan1-1.5')
xlabel('Real part'), grid
```

```
subplot(212)
plot(x,imag(psi))
xlabel('Imaginary part'), grid
```

参考：waveinfo

15．symaux

功能：计算 Symlet 小波

语法：

```
W = SYMAUX(N,SUMW)
W = SYMAUX(N)
```

举例：

```
wdb4 = dbaux(4)
wdb4 =
  Columns 1 through 7
    0.1629    0.5055    0.4461   -0.0198   -0.1323    0.0218    0.0233
  Column 8
   -0.0075
P = conv(wrev(wdb4),wdb4)*2;
wsym4 = symaux(4)
wsym4 =
  Columns 1 through 7
    0.0228   -0.0089   -0.0702    0.2106    0.5683    0.3519   -0.0210
  Column 8
   -0.0536
Psym = conv(wrev(wsym4),wsym4)*2;
err = norm(P-Psym)
err =
  7.4988e-016
```

参考：symwavf, wfilters

16．symwavf

功能：Symlet 小波滤波器

语法：

```
F = symwavf(W)
```

举例：

```
w = symwavf('sym4')
w =
 Columns 1 through 7
    0.0228 -0.0089 -0.0702 0.2106 0.5683 0.3519 -0.0210
 Column 8
    -0.0536
```

参考：symaux, waveinfo

D.3 一维连续小波变换

cwt

功能：一维连续小波变换

语法：

```
COEFS = cwt(S,SCALES,'wname')
COEFS = cwt(S,SCALES,'wname','plot')
COEFS = cwt(S,SCALES,'wname',PLOTMODE)
COEFS = cwt(S,SCALES,'wname',PLOTMODE,XLIM)
```

举例：

```
load vonkoch
vonkoch=vonkoch(1:510);
lv = length(vonkoch);
subplot(311), plot(vonkoch);title('Analyzed signal.');
set(gca,'Xlim',[0 510])
[c,l] = wavedec(vonkoch,5,'sym2');
cfd = zeros(5,lv);
for k = 1:5
      d = detcoef(c,l,k);
      d = d(ones(1,2^k),:);
      cfd(k,:) = wkeep(d(:)',lv);
end

cfd = cfd(:);
I = find(abs(cfd)<sqrt(eps));
cfd(I)=zeros(size(I));
cfd = reshape(cfd,5,lv);
subplot(312), colormap(pink(64));
img = image(flipud(wcodemat(cfd,64,'row')));
set(get(img,'parent'),'YtickLabel',[]);
title('Discrete Transform, absolute coefficients.')
ylabel('level')
subplot(313)
ccfs = cwt(vonkoch,1:32,'sym2','plot');
title('Continuous Transform, absolute coefficients.')
colormap(pink(64));
ylabel('Scale')
```

参考：wavedec, wavefun, waveinfo, wcodemat

D.4 一维离散小波变换

1．appcoef

功能： 提取一维近似分量系数

语法：

```
A = appcoef(C,L,'wname',N)
A = appcoef(C,L,'wname')
A = appcoef(C,L,Lo_R,Hi_R)
A = appcoef(C,L,Lo_R,Hi_R,N)
```

举例：

```
load leleccum; s = leleccum(1:3920);
[c,l] = wavedec(s,3,'db1');
ca3 = appcoef(c,l,'db1',3);
```

参考： detcoef, wavedec

2．detcoef

功能： 提取一维细节分量系数

语法：

```
D = detcoef(C,L,N)
D = detcoef(C,L)
```

举例：

```
load leleccum;
s = leleccum(1:3920);
[c,l] = wavedec(s,3,'db1');
[cd1,cd2,cd3] = detcoef(c,l,[1 2 3]);
```

参考： appcoef, wavedec

3．dwt

功能： 一维离散小波变换

语法：

```
[cA,cD] = dwt(X,'wname')
[cA,cD] = dwt(X,'wname','mode',MODE)
[cA,cD] = dwt(X,Lo_D,Hi_D)
[cA,cD] = dwt(X,Lo_D,Hi_D,'mode',MODE)
```

举例：

```
randn('seed',531316785)
s = 2 + kron(ones(1,8),[1 -1]) + ...
    ((1:16).^2)/32 + 0.2*randn(1,16);

[ca1,cd1] = dwt(s,'haar');
subplot(311); plot(s); title('Original signal');
subplot(323); plot(ca1); title('Approx. coef. for haar');
```

```
subplot(324); plot(cd1); title('Detail coef. for haar'),
[Lo_D,Hi_D] = wfilters('haar','d');
[ca1,cd1] = dwt(s,Lo_D,Hi_D);
[ca2,cd2] = dwt(s,'db2');
subplot(325); plot(ca2); title('Approx. coef. for db2');
subplot(326); plot(cd2); title('Detail coef. for db2');
```

参考：dwtmode, idwt, wavedec, waveinfo

4．dwtmode

功能：设置一维离散小波变换模式

语法：

```
ST = dwtmode
dwtmode('mode')
```

举例：

```
clear global
dwtmode
dwtmode('per')
```

参考：idwt, idwt2, dwt, dwt2, wextend

5．idwt

功能：一维离散小波逆变换

语法：

```
X = idwt(cA,cD,'wname')
X = idwt(cA,cD,Lo_R,Hi_R)
X = idwt(cA,cD,'wname',L)
X = idwt(cA,cD,Lo_R,Hi_R,L)
X = idwt(...,'mode',MODE)
```

举例：

```
randn('seed',531316785)
s = 2 + kron(ones(1,8),[1 -1]) + ...
    ((1:16).^2)/32 + 0.2*randn(1,16);
[ca1,cd1] = dwt(s,'db2');
subplot(221); plot(ca1);
title('Approx. coef. for db2');
subplot(222); plot(cd1);
title('Detail coef. for db2');
ss = idwt(ca1,cd1,'db2');
err = norm(s-ss);        % Check reconstruction.
subplot(212); plot([s;ss]');
title('Original and reconstructed signals');
xlabel(['Error norm = ',num2str(err)])
[Lo_R,Hi_R] = wfilters('db2','r');
ss = idwt(ca1,cd1,Lo_R,Hi_R);
```

参考：dwt, dwtmode, upwlev

6．upcoef

功能：由一维离散小波变换求解近似和细节分量

语法：

```
Y = upcoef(O,X,'wname',N)
Y = upcoef(O,X,'wname',N,L)
Y = upcoef(O,X,Lo_R,Hi_R,N)
Y = upcoef(O,X,Lo_R,Hi_R,N,L)
Y = upcoef(O,X,'wname')
Y = upcoef(O,X,Lo_R,Hi_R)
```

举例：

```
cfs = [1];   % Decomposition reduced a single coefficient.
essup = 10; % Essential support of the scaling filter db6.
figure(1)
for i=1:6
    rec = upcoef('a',cfs,'db6',i);
    ax = subplot(6,1,i),h = plot(rec(1:essup));
    set(ax,'xlim',[1 325]);
    essup = essup*2;
end
subplot(611)
title(['Approximation signals, obtained from a single  ...
      'coefficient at levels 1 to 6'])
```

参考：idwt

7．upwlev

功能：单层小波分解的重构

语法：

```
[NC,NL,cA] = upwlev(C,L,'wname')
[NC,NL,cA] = upwlev(C,L,Lo_R,Hi_R)
```

举例：

```
load sumsin; s = sumsin;
[c,l] = wavedec(s,3,'db1');
subplot(311); plot(s);
title('Original signal s.');
subplot(312); plot(c);
title('Wavelet decomposition structure, level 3')
xlabel(['Coefs for approx. at level 3 ' ...
        'and for det. at levels 3, 2 and 1'])
[nc,nl] = upwlev(c,l,'db1');
subplot(313); plot(nc);
title('Wavelet decomposition structure, level 2')
xlabel(['Coefs for approx. at level 2 ' ...
        'and for det. at levels 2 and 1'])
```

参考：idwt, upcoef, wavedec

8. wavedec

功能：多层一维小波分解

语法：

```
[C,L] = wavedec(X,N,'wname')
[C,L] = wavedec(X,N,Lo_D,Hi_D)
```

举例：

```
load sumsin; s = sumsin;
[c,l] = wavedec(s,3,'db1');
```

参考：dwt, waveinfo, waverec, wfilters, wmaxlev

9. waverec

功能：多层一维小波重构

语法：

```
X = waverec(C,L,'wname')
X = waverec(C,L,Lo_R,Hi_R)
```

举例：

```
I load leleccum; s = leleccum(1:3920); ls = length(s);
[c,l] = wavedec(s,3,'db5');
a0 = waverec(c,l,'db5');
err = norm(s-a0)
err =
     3.2079e-09
```

参考：appcoef, idwt, wavedec

10. wenergy

功能：小波或小波包分解能量函数

语法：

```
[Ea,Ed] = WENERGY(C,L)
E = WENERGY(T)
```

举例：

```
load noisbump
[C,L] = wavedec(noisbump,4,'sym4');
[Ea,Ed] = wenergy(C,L)
Ea =
   88.2860
Ed =
      2.1560      1.2286      1.4664      6.8630
```

参考：无

11. wrcoef

功能：一维信号小波重构

语法：

```
X = wrcoef('type',C,L,'wname',N)
X = wrcoef('type',C,L,Lo_R,Hi_R,N)
```

```
X = wrcoef('type',C,L,'wname')
X = wrcoef('type',C,L,Lo_R,Hi_R)
freqz2(...)
```

举例:

```
load sumsin; s = sumsin;
[c,l] = wavedec(s,5,'sym4');
a5 = wrcoef('a',c,l,'sym4',5);
```

参考: appcoef, detcoef, wavedec

D.5　二维离散小波变换

1. appcoef2

功能：提取二维离散小波变换的近似分量

语法:

```
A = appcoef2(C,S,'wname',N)
A = appcoef2(C,S,'wname')
A = appcoef2(C,S,Lo_R,Hi_R)
A = appcoef2(C,S,Lo_R,Hi_R,N)
```

举例:

```
load woman;
[c,s] = wavedec2(X,2,'db1');
sizex = size(X)
sizex =
    256 256
sizec = size(c)
sizec =
      1       65536
    val_s = s
val_s =
    64      64
    64      64
   128     128
   256     256
ca2 = appcoef2(c,s,'db1',2);
sizeca2 = size(ca2)
sizeca2 =
    64      64
ca1 = appcoef2(c,s,'db1',1);
sizeca1 = size(ca1)
sizeca1 =
   128     128
```

参考: detcoef2, wavedec2

2．detcoef2

功能：提取二维离散小波变换的细节分量

语法：

```
D = detcoef2(O,C,S,N)
```

举例：

```
load woman;
[c,s] = wavedec2(X,2,'db1');
sizex = size(X)
sizex =
     256      256
sizec = size(c)
sizec =
       1     65536
val_s = s
val_s =
      64       64
      64       64
     128      128
     256      256
[chd2,cvd2,cdd2] = detcoef2('all',c,s,2);
sizecd2 = size(chd2)
sizecd2 =
      64       64
[chd1,cvd1,cdd1] = detcoef2('all',c,s,1);
sizecd1 = size(chd1)
sizecd1 =
     128      128
```

参考：appcoef2, wavedec2

3．dwt2

功能：单层二维离散小波变换

语法：

```
[cA,cH,cV,cD] = dwt2(X,'wname')
[cA,cH,cV,cD] = dwt2(X,Lo_D,Hi_D)
```

举例：

```
load woman;
nbcol = size(map,1);
[cA1,cH1,cV1,cD1] = dwt2(X,'db1');
cod_X = wcodemat(X,nbcol);
cod_cA1 = wcodemat(cA1,nbcol);
cod_cH1 = wcodemat(cH1,nbcol);
cod_cV1 = wcodemat(cV1,nbcol);
cod_cD1 = wcodemat(cD1,nbcol);
dec2d = [...
          cod_cA1,     cod_cH1;      ...
```

```
            cod_cV1,      cod_cD1      ...
            ];
```

参考：dwtmode, idwt2, wavedec2, waveinfo

4．idwt2

功能：单层二维离散小波逆变换

语法：

```
X = idwt2(cA,cH,cV,cD,'wname')
X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R)
X = idwt2(cA,cH,cV,cD,'wname',S)
X = idwt2(cA,cH,cV,cD,Lo_R,Hi_R,S)
X = idwt2(...,'mode',MODE)
```

举例：

```
load woman;
sX = size(X);
[cA1,cH1,cV1,cD1] = dwt2(X,'db4');
A0 = idwt2(cA1,cH1,cV1,cD1,'db4',sX);
max(max(abs(X-A0)))
ans =
    3.4176e-10
```

参考：dwt2, dwtmode, upwlev2

5．upcoef2

功能：由多层小波分解重构近似分量或细节分量

语法：

```
Y = upcoef2(O,X,'wname',N,S)
Y = upcoef2(O,X,Lo_R,Hi_R,N,S)
Y = upcoef2(O,X,'wname',N)
Y = upcoef2(O,X,Lo_R,Hi_R,N)
Y = upcoef2(O,X,'wname')
Y = upcoef2(O,X,Lo_R,Hi_R)
```

举例：

```
load woman;
[c,s] = wavedec2(X,2,'db4');
siz = s(size(s,1),:);
ca1 = appcoef2(c,s,'db4',1);
a1 = upcoef2('a',ca1,'db4',1,siz);
chd1 = detcoef2('h',c,s,1);
hd1 = upcoef2('h',chd1,'db4',1,siz);
cvd1 = detcoef2('v',c,s,1);
vd1 = upcoef2('v',cvd1,'db4',1,siz);
cdd1 = detcoef2('d',c,s,1);
dd1 = upcoef2('d',cdd1,'db4',1,siz);
```

参考：idwt2

6．upwlev2

功能：二维信号小波分解的单层重构

语法：

```
[NC,NS,cA] = upwlev2(C,S,'wname')
[NC,NS,cA] = upwlev2(C,S,Lo_R,Hi_R)
```

举例：

```
load woman;
[c,s] = wavedec2(X,2,'db1');
sc = size(c)
sc =
     1    65536
val_s = s
val_s =
     64    64
     64    64
     128   128
     256   256
[nc,ns] = upwlev2(c,s,'db1');
snc = size(nc)
snc =
     1    65536
val_ns = ns
val_ns =
     128    128
     128    128
     256    256
```

参考：idwt2, upcoef2, wavedec2

7．wavedec2

功能：二维信号的多层小波分解

语法：

```
[C,S] = wavedec2(X,N,'wname')
[C,S] = wavedec2(X,N,Lo_D,Hi_D)
```

举例：

```
load woman;
[c,s] = wavedec2(X,2,'db1');
sizex = size(X)
sizex =
     256   256
sizec = size(c)
sizec =
     1   65536
     val_s = s
val_s =
     64   64
```

```
64   64
128   128
256   256
```

参考：dwt, waveinfo, waverec2, wfilters, wmaxlev

8．waverec2

功能：二维信号多层小波重构

语法：

```
X = waverec2(C,S,'wname')
X = waverec2(C,S,Lo_R,Hi_R)
```

举例：

```
load woman;
[c,s] = wavedec2(X,2,'sym4');
a0 = waverec2(c,s,'sym4');
max(max(abs(X-a0)))
ans =
    2.5565e-10
```

参考：fft, fft2, fftn, ifftn

9．wenergy2

功能：计算二维小波分解能量

语法：

```
[Ea,Eh,Ev,Ed] = WENERGY2(C,S)
[Ea,EDetail] = WENERGY2(C,S)
```

举例：

```
load detail
[C,S] = wavedec2(X,2,'sym4');
[Ea,Eh,Ev,Ed] = wenergy2(C,S)
Ea =
   89.3520
Eh =
    1.8748      2.7360
Ev =
    1.5860      2.6042
Ed =
    0.7539      1.0932
[Ea,EDetails] = wenergy2(C,S)
Ea =
   89.3520
EDetails =
    4.2147      6.4334
```

参考：无

D.6 小波包算法

1．bestlevt

功能：利用小波包分解寻找最优树

语法：

```
T = bestlevt(T)
[T,E] = bestlevt(T)
```

举例：

```
wpt = wpdec(x,3,'db1');
blt = bestlevt(wpt);
```

参考：besttree, wenergy, wpdec, wpdec2

2．besttree

功能：利用小波包分解寻找最优层次树

语法：

```
T = besttree(T)
[T,E] = besttree(T)
[T,E,N] = besttree(T)
```

举例：

```
wpt = wpdec(x,3,'db1');
wpt = wpsplt(wpt,[3 0]);
bt = besttree(wpt);
```

参考：bestlevt, wenergy, wpdec, wpdec2

3．entrupd

功能：按新的熵函数更新分解结构

语法：

```
T = entrupd(T,ENT)
T = entrupd(T,ENT,PAR)
```

举例：

```
t = wpdec(x,2,'db1','shannon');
t = entrupd(t,'threshold',0.5);
```

参考：wenergy, wpdec, wpdec2

4．wentropy

功能：计算熵

语法：

```
E = wentropy(X,T,P)
E = wentropy(X,T)
```

举例：

```
x = randn(1,200);
```

```
e = wentropy(x,'shannon')
```

参考：无

5．wp2wtree

功能：从小波包分解树中提取小波分解树

语法：

```
T = wp2wtree(T)
```

举例：

```
wpt = wpdec(x,3,'db1');
wt = wp2wtree(wpt);
```

参考：wpdec, wpdec2

6．wpcoef

功能：按指定节点的小波包分解

语法：

```
X = wpcoef(T,N)
X = wpcoef(T)
```

举例：

```
wpt = wpdec(x,3,'db1');
cfs = wpcoef(wpt,[2 1]);
```

参考：wpdec, wpdec2

7．wpcutree

功能：对小波包分解树进行剪切

语法：

```
T = wpcutree(T,L)
[T,RN] = wpcutree(T,L)
```

举例：

```
wpt = wpdec(x,3,'db1');
nwpt = wpcutree(wpt,2);
```

参考：wpdec, wpdec2

8．wpdec

功能：小波包的一维分解

语法：

```
T = wpdec(X,N,'wname',E,P)
T = wpdec(X,N,'wname')
```

举例：

```
load noisdopp; x = noisdopp;
wpt = wpdec(x,3,'db1','shannon');
```

参考：wavedec, waveinfo, wenergy, wpdec, wprec

9．wpdec2

功能：小波包的二维分解

语法：

```
T = wpdec2(X,N,'wname',E,P)
T = wpdec2(X,N,'wname')
```

举例：

```
load tire
t = wpdec2(X,2,'db1');
```

参考： wavedec2, waveinfo, wenergy, wpdec, wprec2

10. wpfun

功能： 小波包函数簇

语法：

```
[WPWS,X] = wpfun('wname',NUM,PREC)
[WPWS,X] = wpfun('wname',NUM)
```

举例：

```
[wp,x] = wpfun('db2',7);
```

参考： wavefun, waveinfo

11. wpjoin

功能： 重组小波包

语法：

```
T = wpjoin(T,N)
[T,X] = wpjoin(T,N)
T = wpjoin(T)
[T,X] = wpjoin(T)
```

举例：

```
wpt = wpdec(x,3,'db1');
wpt = wpjoin(wpt,[1 1]);
```

参考： wpdec, wpdec2, wpsplt

12. wprcoef

功能： 重构小波包分解树的系数

语法：

```
X = wprcoef(T,N)
```

举例：

```
load noisdopp; x = noisdopp;
t = wpdec(x,3,'db1','shannon');
rcfs = wprcoef(t,[2 1]);
```

参考： wpdec, wpdec2, wprec, wprec2

13. wprec

功能： 小波包的一维重构

语法：

```
X = wprec(T)
```

举例：

无

参考：wpdec, wpdec2, wpjoin, wprec2, wpsplt

14. wprec2

功能：小波包的二维重构

语法：

```
X = wprec2(T)
```

举例：

无

参考：wpdec, wpdec2, wpjoin, wprec, wpsplt

15. wpsplt

功能：分裂小波包分解的终端节点

语法：

```
T = wpsplt(T,N)
[T,cA,cD] = wpsplt(T,N)
[T,cA,cH,cV,cD] = wpsplt(T,N)
```

举例：

```
load noisdopp;
x = noisdopp;
wpt = wpdec(x,3,'db1');
wpt = wpsplt(wpt,[3 0]);
```

参考：无

D.7 离散静态小波变换

1. iswt

功能：一维离散静态小波逆变换

语法：

```
X = iswt(SWC,'wname')
X = iswt(SWA,SWD,'wname')
X = iswt(SWC,Lo_R,Hi_R)
X = iswt(SWA,SWD,Lo_R,Hi_R)
```

举例：

```
load noisbloc; s = noisbloc;
swc = swt(s,3,'db1');
a0 = iswt(swc,'db1');
```

参考：idwt, swt, waverec

2. iswt2

功能：二维离散静态小波逆变换

语法：

```
X = iswt2(SWC,'wname')
X = iswt2(A,H,V,D,'wname')
X = iswt2(SWC,Lo_R,Hi_R)
X = iswt2(A,H,V,D,Lo_R,Hi_R)
```

举例：

```
load nbarb1;
swc = swt2(X,3,'sym4');
a0 = iswt2(swc,'sym4');
```

参考：idwt2, swt2, waverec2

3．swt

功能：一维离散静态小波变换

语法：

```
SWC = swt(X,N,'wname')
SWC = swt(X,N,Lo_D,Hi_D)
[SWA,SWD] = swt(X,N,'wname')
[SWA,SWD] = swt(X,N,Lo_D,Hi_D)
```

举例：

```
load noisbloc; s = noisbloc;
[swa,swd] = swt(s,3,'db1');
```

参考：dwt, wavedec

4．swt2

功能：二维离散静态小波变换

语法：

```
SWC = swt2(X,N,'wname')
[A,H,V,D] = swt2(X,N,'wname')
SWC = swt2(X,N,Lo_D,Hi_D)
[A,H,V,D] = swt2(X,N,Lo_D,Hi_D)
```

举例：

```
load nbarb1;
nbcol = size(map,1);
cod_X = wcodemat(X,nbcol);
[ca,chd,cvd,cdd] = swt2(X,3,'sym4');
```

参考：dwt2, wavedec2

D.8 信号/图像的去噪和压缩函数

1．ddencmp

功能：自动生成小波去噪或压缩阈值选择方案

语法：

```
[THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,IN2,X)
[THR,SORH,KEEPAPP] = ddencmp(IN1,'wv',X)
```

```
[THR,SORH,KEEPAPP,CRIT] = ddencmp(IN1,'wp',X)
```

举例:

```
init = 2055415866; randn('seed',init);
x = randn(1,1000);
[thr,sorh,keepapp] = ddencmp('den','wv',x)
```

参考: wdencmp, wenergy, wpdencmp

2. thselect

功能: 为利用小波分解去噪选取阈值

语法:

```
THR = thselect(X,TPTR)
```

举例:

```
init = 2055415866; randn('seed',init);
x = randn(1,1000);
thr = thselect(x,'rigrsure')
```

参考: wden

3. wbmpen

功能: 根据惩罚函数对小波的一维或二维去噪选择阈值

语法:

```
THR = wbmpen(C,L,SIGMA,ALPHA)
```

举例:

```
load noisbump; x = noisbump;
wname = 'sym6'; lev = 5;
[c,l] = wavedec(x,lev,wname);
sigma = wnoisest(c,l,1);
alpha = 2;
thr = wbmpen(c,l,sigma,alpha)
```

参考: wden, wdencmp, wpbmpen, wpdencmp

4. wdcbm

功能: 利用 Birge-Massart 准则进行一维小波去噪

语法:

```
[THR,NKEEP] = wdcbm(C,L,ALPHA)
[THR,NKEEP] = wdcbm(C,L,ALPHA,M)
```

举例:

```
x = leleccum(indx);
wname = 'db3'; lev = 5;
[c,l] = wavedec(x,lev,wname);
alpha = 1.5; m = l(1);
[thr,nkeep] = wdcbm(c,l,alpha,m)
```

参考: wden, wdencmp, wpdencmp

5. wdcbm2

功能: 利用 Birge-Massart 准则进行二维小波去噪

语法:

```
[THR,NKEEP] = wdcbm2(C,S,ALPHA)
[THR,NKEEP] = wdcbm2(C,S,ALPHA,M)
```

举例:

```
load detfingr;
nbc = size(map,1);
wname = 'sym4'; lev = 3;
[c,s] = wavedec2(X,lev,wname);
alpha = 1.5; m = 2.7*prod(s(1,:));
[thr,nkeep] = wdcbm2(c,s,alpha,m)
```

参考: wdencmp, wpdencmp

6. wden

功能: 自动利用小波进行一维信号去噪

语法:

```
[XD,CXD,LXD] = wden(X,TPTR,SORH,SCAL,N,'wname')
[XD,CXD,LXD] = wden(C,L,TPTR,SORH,SCAL,N,'wname')
```

举例:

```
[c,l] = wavedec(x,lev,'sym8');
xd = wden(c,l,'minimaxi','s','sln',lev,'sym8');
```

参考: thselect, wavedec, wdencmp, wfilters, wthresh

7. wdencmp

功能: 利用小波对信号进行去噪或压缩

语法:

```
[XC,CXC,LXC,PERF0,PERFL2] =
wdencmp('gbl',X,'wname',N,THR,SORH,KEEPAPP)
[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('lvd',X,'wname',N,THR,SORH)
[XC,CXC,LXC,PERF0,PERFL2] = wdencmp('lvd',C,L,'wname',N,THR,SORH)
```

举例:

```
init=2055615866; randn('seed',init);
x = X + 18*randn(size(X));
[thr,sorh,keepapp] = ddencmp('den','wv',x);
xd = wdencmp('gbl',x,'sym4',2,thr,sorh,keepapp);
```

参考: ddencmp, wavedec, wavedec2, wdcbm, wdcbm2, wden, wbmpen, wpdencmp, thresh

8. wnoise

功能: 为检验小波去噪效果产生测试噪声

语法:

```
X = wnoise(FUN,N)
[X,XN] = wnoise(FUN,N,SQRT_SNR)
[X,XN] = wnoise(FUN,N,SQRT_SNR,INIT)
```

举例:

```
x = wnoise(3,10);
```

参考: wden

9. wnoisest

功能：估计一维小波系数中噪声的能量

语法：

```
STDC = wnoisest(C,L,S)
```

举例：

```
init = 2055415866; randn('seed',init);
x = randn(1,1000);
[c,l] = wavedec(x,2,'db3');
wnoisest(c,l,1:2)
```

参考：thselect, wavedec, wden

10. wpbmpen

功能：利用 Birge-Massart 准则进行小波包去噪

语法：

```
THR = wpbmpen(T,SIGMA,ALPHA)
THR = wpbmpen(T,SIGMA,ALPHA,ARG)
```

举例：

```
load noischir; x = noischir;
wname = 'sym6'; lev = 5;
tree = wpdec(x,lev,wname);
det1 = wpcoef(tree,2);
sigma = median(abs(det1))/0.6745;
alpha = 2;
thr = wpbmpen(tree,sigma,alpha)
```

参考：wbmpen, wden, wdencmp, wpdencmp

11. wpdencmp

功能：利用小波包分解进行信号的压缩或去噪

语法：

```
[XD,TREED,PERF0,PERFL2] =
wpdencmp(X,SORH,N,'wname',CRIT,PAR,KEEPAPP)
[XD,TREED,PERF0,PERFL2] =wpdencmp(TREE,SORH,CRIT,PAR,KEEPAPP)
```

举例：

```
[xc,treed,datad,perf0,perfl2] = wpdencmp(x,sorh,3,'db2',crit,thr,keepapp);
```

参考：besttree, ddencmp, wdencmp, wenergy, wpbmpen, wpdec, wpdec2, wthresh

12. wpthcoef

功能：对小波包分解系数进行阈值处理

语法：

```
T = wpthcoef(T,KEEPAPP,SORH,THR)
```

举例：

无

参考：wpdec, wpdec2, wpdencmp, wthresh

13．wthcoef

功能：对一维小波分解系数进行阈值处理

语法：

```
NC = wthcoef('d',C,L,N,P)
NC = wthcoef('d',C,L,N)
NC = wthcoef('a',C,L)
NC = wthcoef('t',C,L,N,T,SORH)
```

举例：

无

参考：wavedec, wthresh

14．wthcoef2

功能：对二维小波分解系数进行阈值处理

语法：

```
NC = wthcoef2('type',C,S,N,T,SORH)
NC = wthcoef2('type',C,S,N)
NC = wthcoef2('a',C,S)
NC = wthcoef2('t',C,S,N,T,SORH)
```

举例：

无

参考：wavedec2, wthresh

15．wthresh

功能：进行软阈值或硬阈值处理

语法：

```
Y = wthresh(X,SORH,T)
```

举例：

```
y = linspace(-1,1,100);
thr = 0.4;
ythard = wthresh(y,'h',thr);
ytsoft = wthresh(y,'s',thr);
```

参考：wden, wdencmp, wpdencmp

16．wthrmngr

功能：阈值设置管理

语法：

```
THR = wthrmngr(OPTION,METHOD,VARARGIN)
```

举例：

```
THR = wthrmngr('wp2ddenoGBL','sqtwologuwn',WPT)
```

参考：无

参考文献

[1] 秦前清，杨宗凯. 实用小波分析. 西安：西安电子科技大学出版社，1994
[2] 胡昌华，张军波等. 基于 MATLAB 的系统分析与设计——小波分析. 西安：西安电子科技大学出版社，1999
[3] 程正兴. 小波分析算法与应用. 西安：西安交通大学出版社，1998
[4] 杨福生. 小波变换的工程分析与应用. 北京：科学出版社，1999
[5] 彭玉华. 小波变换与工程应用. 北京：科学出版社，1999
[6] Wavelet Toolbox User's Guide. Mathworks Inc，2002
[7] The MathWorks，Inc. http://www.mathworks.com，2002